Engaging Data Visualizations for Graphical Storytelling

From Theory to Practice with ggplot2

Dr Cédric Scherer
Fannie Mae | September 2024

Setup

The Data Set(s)

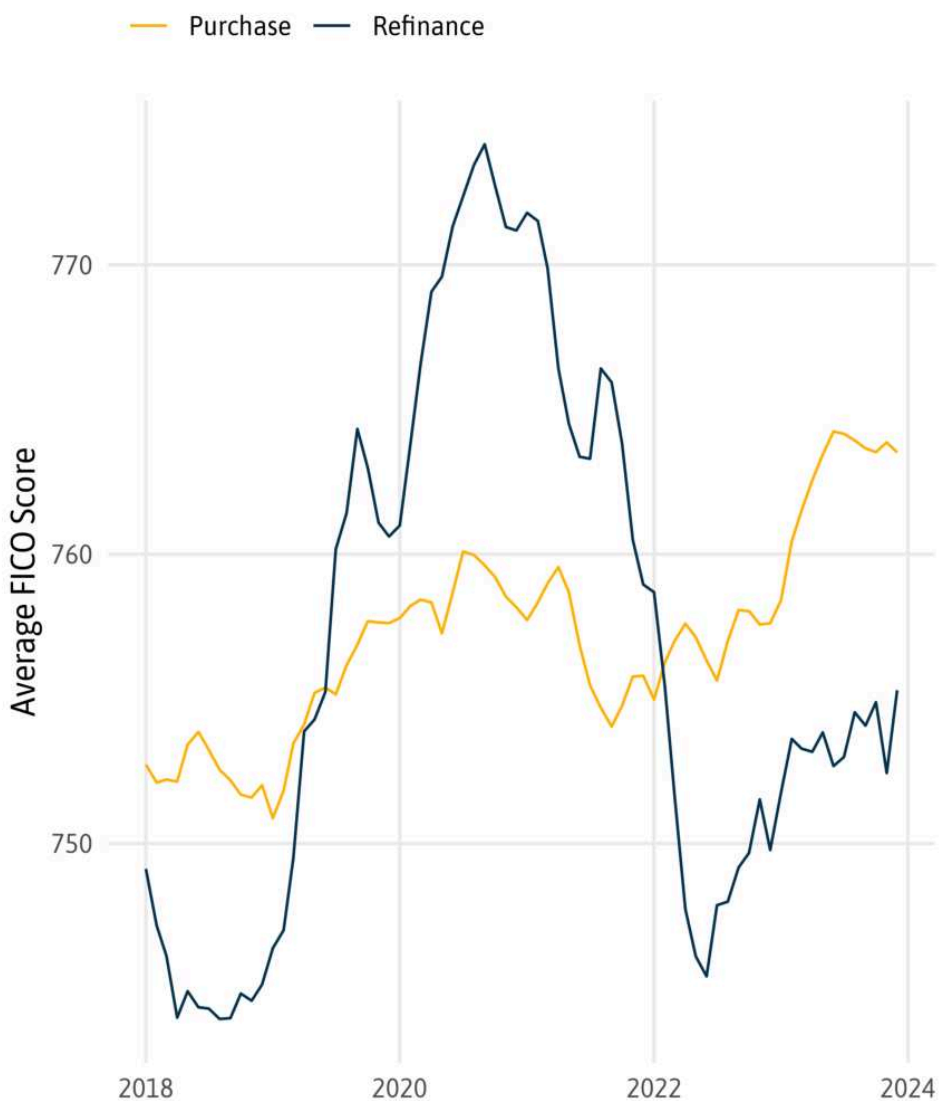
```r
library(dplyr)
library(lubridate)

fannie_raw <-
  readr::read_csv(
    here::here("data", "fannie-mae-data-18-23.csv"),
    col_types = "Dddddfccfd"
  ) |>
  filter(purpose %in% c("R", "P")) |>
  mutate(purpose = if_else(purpose == "R", "Refinance", "Purchase"))

set.seed(2024)
fannie_sub <- slice_sample(fannie_raw, n = 2000)

fannie_agg <-
  fannie_raw |>
  summarize(
    loans = n(),
    avg_rate = mean(orig_rate, na.rm = TRUE),
    avg_score = mean(cscore_b, na.rm = TRUE),
    avg_upb_dti = sum(dti * orig_upb, na.rm = TRUE) / sum(orig_upb, na.rm = TRUE),
    .by = c(orig_date, year, purpose)
  )
```

Prerequisites

```
library(ggplot2)

theme_set(
  theme_minimal(
    base_size = 13,
    base_family = "Asap SemiCondensed"
  )
)

theme_update(
  panel.grid.minor = element_blank(),
  plot.title.position = "plot",
  legend.position = "top",
  legend.justification = "left"
)
```

Store a ggplot

```
g <-
  ggplot(
    fannie_agg,
    aes(x = orig_date, y = avg_scor
        color = purpose)
  ) +
  geom_line() +
  scale_color_manual(
    values = c("#FFB306", "#05314D'
    name = NULL
  ) +
  labs(x = NULL, y = "Average FICO

g
```



Working with Fonts

Cédric Scherer Data Visualization & Information Design

The {systemfonts} Package

locates fonts and provides font-related utilities to graphic devices

```
1 library(systemfonts)
```

```
1 match_font("Asap", bold = TRUE)
```

```
$path
[1] "/Users/cedric/Library/Fonts/Asap-Bold.ttf"

$index
[1] 0

$features
integer(0)
```

Use Non-Default Typefaces

```
system_fonts()
```

```
# A tibble: 2,825 × 9
   path                  index name  family style weight width italic monospace
   <chr>                 <int> <chr> <chr>  <chr> <ord>  <ord> <lgl>  <lgl>
 1 /Users/cedric/Library…     0 Glor… Glori… Extr… ultra… norm… FALSE  FALSE
 2 /Users/cedric/Library…     0 Jose… Josef… Extr… normal norm… FALSE  FALSE
 3 /Users/cedric/Library…     0 Lite… Liter… Bold  bold   norm… FALSE  FALSE
 4 /Users/cedric/Library…     0 false false  Norm… normal norm… FALSE  FALSE
 5 /System/Library/Fonts…     0 Beir… Beirut Regu… bold   norm… FALSE  FALSE
 6 /Users/cedric/Library…     0 Bric… Brico… Extr… ultra… semi… FALSE  FALSE
 7 /Users/cedric/Library…     0 Huov… Huova  Regu… normal norm… FALSE  FALSE
 8 /System/Library/Fonts…     0 ITFD… ITF D… Book  normal norm… FALSE  FALSE
 9 /Users/cedric/Library…     0 Mona… Monas… Wide… light  norm… FALSE  FALSE
10 /Users/cedric/Library…     0 Asap… Asap … Thin  normal semi… FALSE  FALSE
# ℹ 2,815 more rows
```

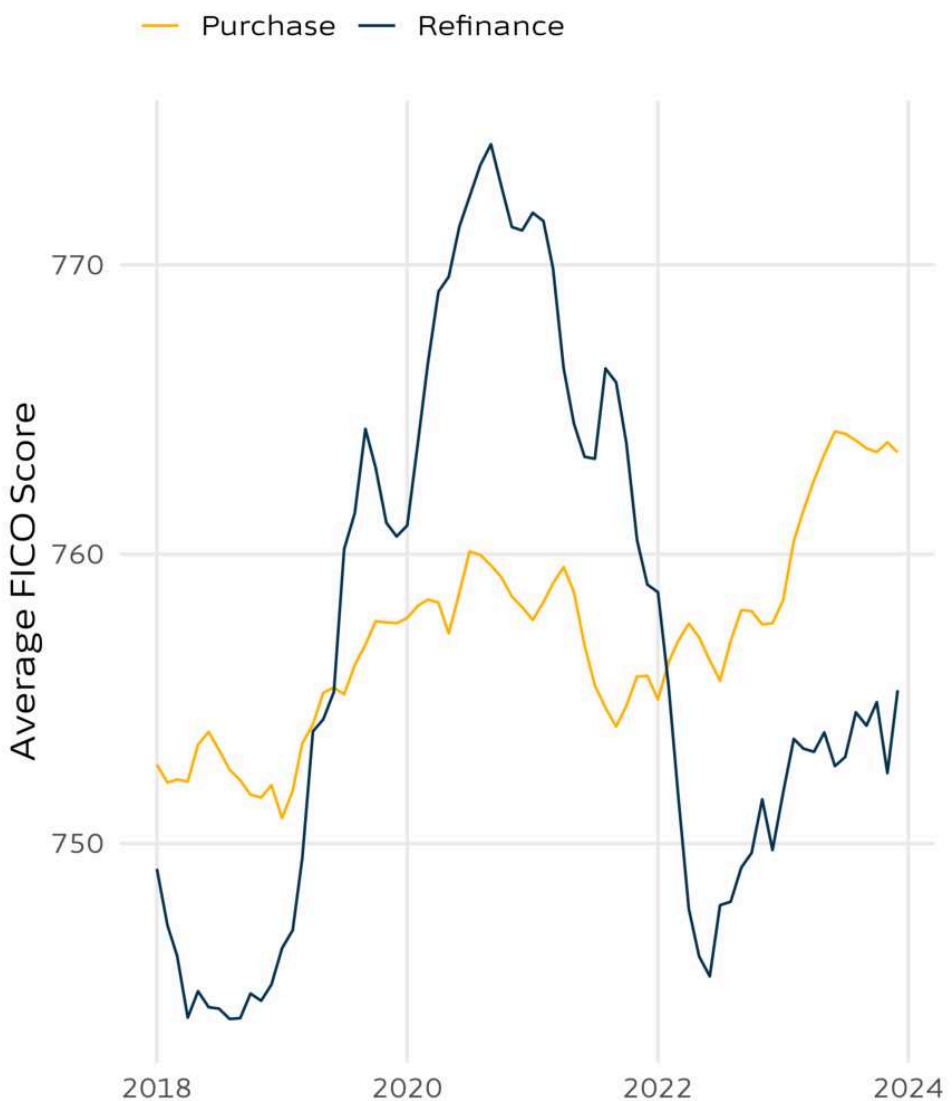
Use Non-Default Typefaces

```
system_fonts() |>
  filter(stringr::str_detect(family, "Asap")) |>
  select(family) |>
  unique() |>
  arrange(family)
```

```
# A tibble: 5 × 1
  family
  <chr>
1 Asap
2 Asap Condensed
3 Asap Expanded
4 Asap SemiCondensed
5 Asap SemiExpanded
```

Use Non-Default Typefaces

```
g +
  theme(
    text = element_text(
      family = "Asap Expanded"
    )
  )
```



Use Font Features

Cédric Scherer Data Visualization & Information Design

Use Font Features

```
system_fonts() |>
  filter(family == "Asap SemiCondensed") |>
  select(name) |>
  arrange(name)
```

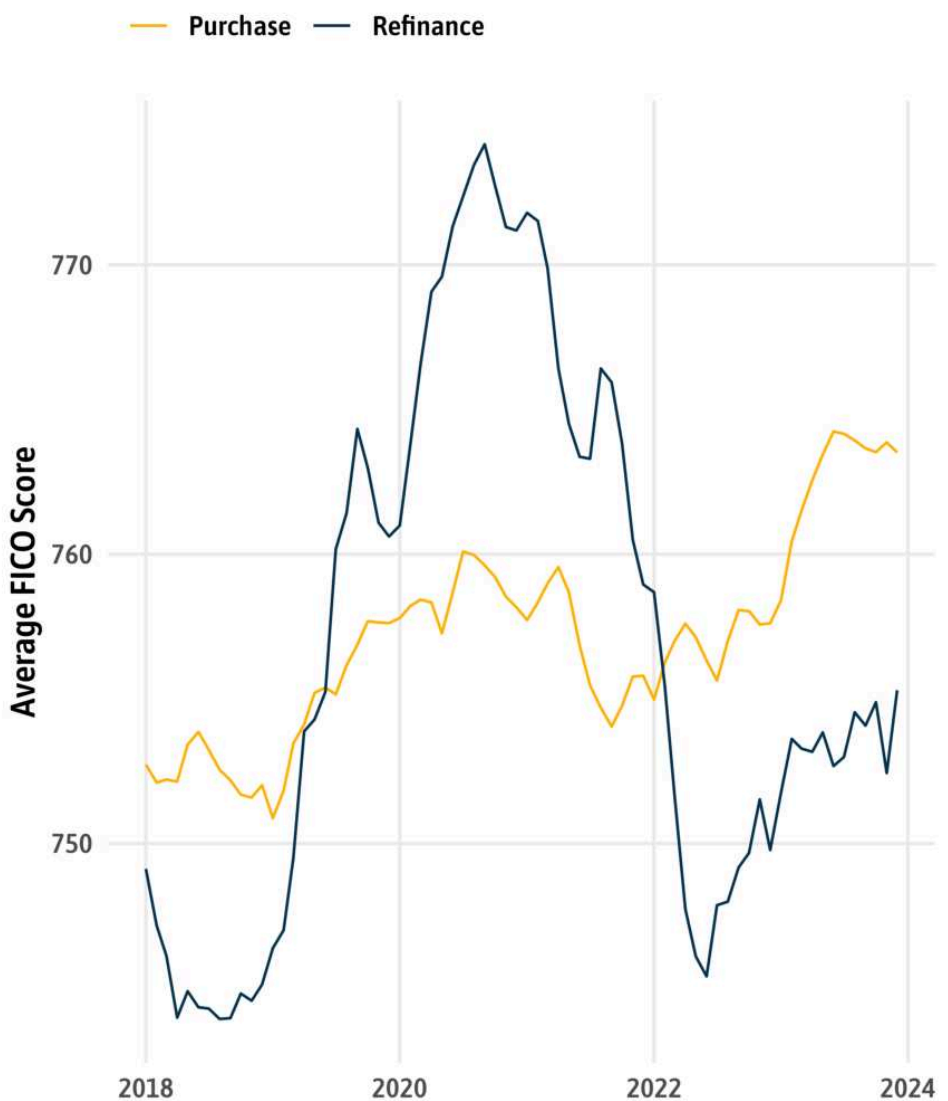
```
# A tibble: 18 × 1
   name
   <chr>
 1 AsapSemiCondensed-Black
 2 AsapSemiCondensed-BlackItalic
 3 AsapSemiCondensed-Bold
 4 AsapSemiCondensed-BoldItalic
 5 AsapSemiCondensed-ExtraBold
 6 AsapSemiCondensed-ExtraBoldItalic
 7 AsapSemiCondensed-ExtraLight
 8 AsapSemiCondensed-ExtraLightItalic
 9 AsapSemiCondensed-Italic
10 AsapSemiCondensed-Light
11 AsapSemiCondensed-LightItalic
12 AsapSemiCondensed-Medium
13 AsapSemiCondensed-MediumItalic
14 AsapSemiCondensed-Regular
15 AsapSemiCondensed-SemiBold
16 AsapSemiCondensed-SemiBoldItalic
```

Use Font Features

```
register_variant(
  name = "Asap SemiCondensed Semibold S1",
  family = "Asap SemiCondensed",
  weight = "semibold",
  features = font_feature(letters = "stylistic")
)
```

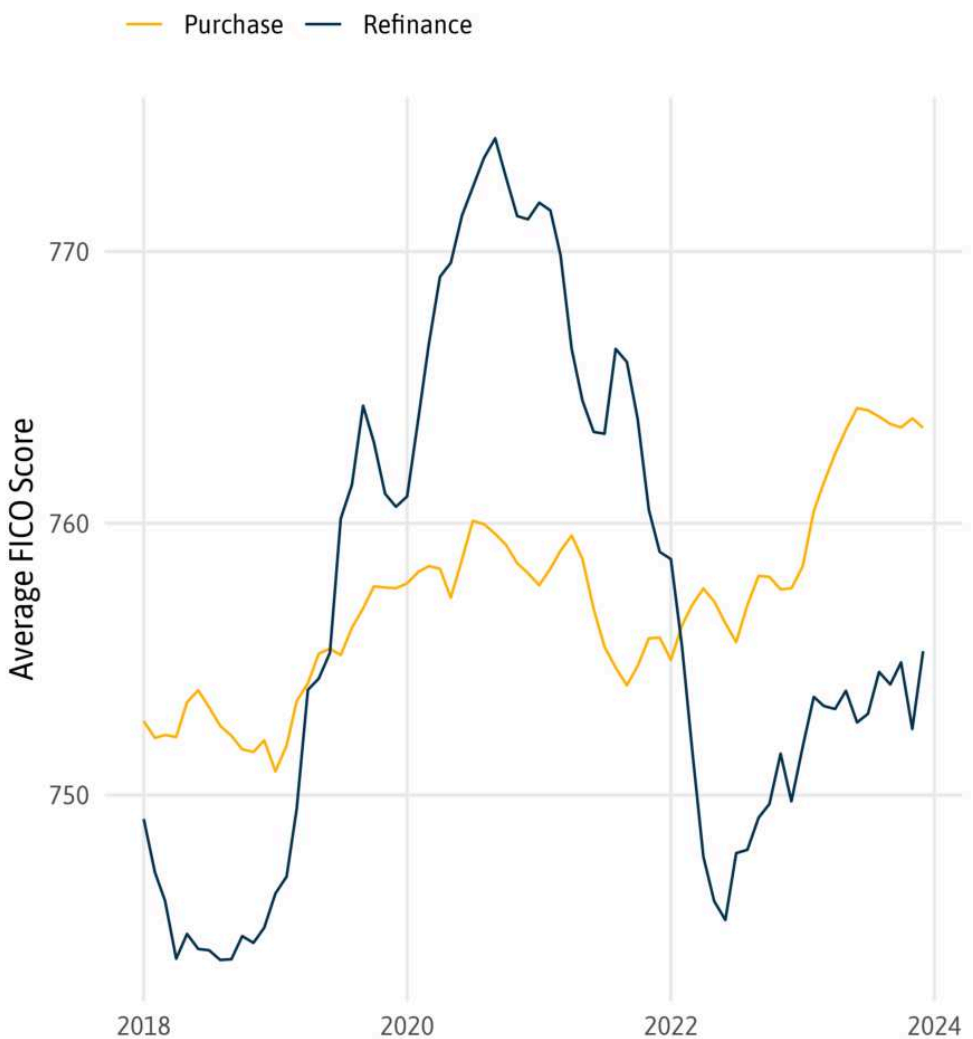
Use Font Features

```
g +
  theme(
    text = element_text(
      family = "Asap SemiCondensed
    )
  )
```

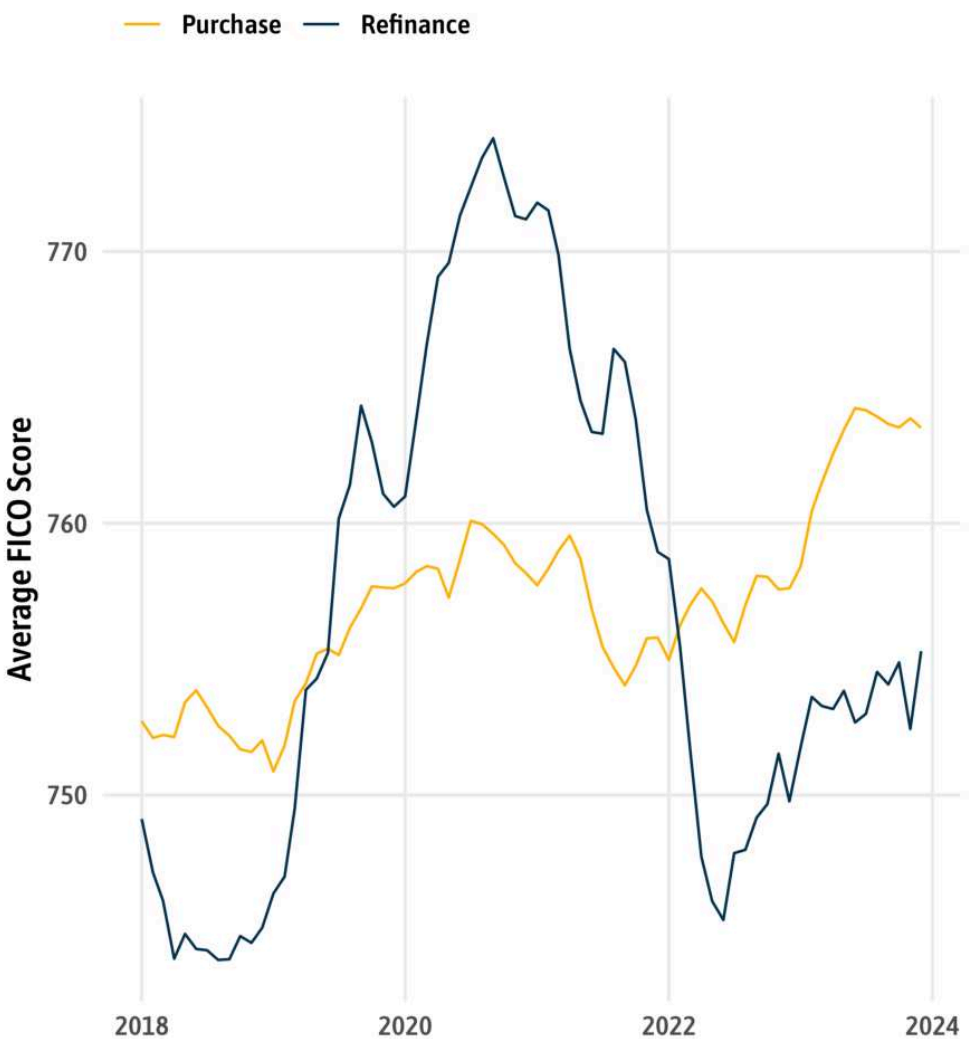


Use Font Features

You're boring... 😪



Yikes, that's exciting! 🤩



Use Font Features

```
register_variant(
  name = "Spline Sans Tabular",
  family = "Spline Sans",
  weight = "normal",
  features = font_feature(numbers = "tabular")
)
```

Use Font Features

Proportional	Tabular
1,509.67	1,509.67
710.13	710.13
498.30	498.30
11.11	11.11
8.57	8.57

The {ragg} Package

provides drop-in replacements for the default raster graphic devices

- faster
- direct access to all system fonts
- advanced text rendering
  - including support for right-to-left text, emojis, and font fallback
- high quality anti-aliasing
- high quality rotated text
- supports 16-bit output
- system independent rendering

The {ragg} Package

Ragg device (macOS)

x ← y ≠ z

Ragg device (Windows)

x ← y ≠ z

Ragg device (Linux)

x ← y ≠ z

Cairo device (macOS)

x <- y != z

twitter

Cairo device (Windows)

x <- y != z

twitter

Cairo device (Linux)

x ← y ≠ z

Quartz device (macOS)

x <- y != z

twitter

Windows device (Windows)

x <- y != z

twitter

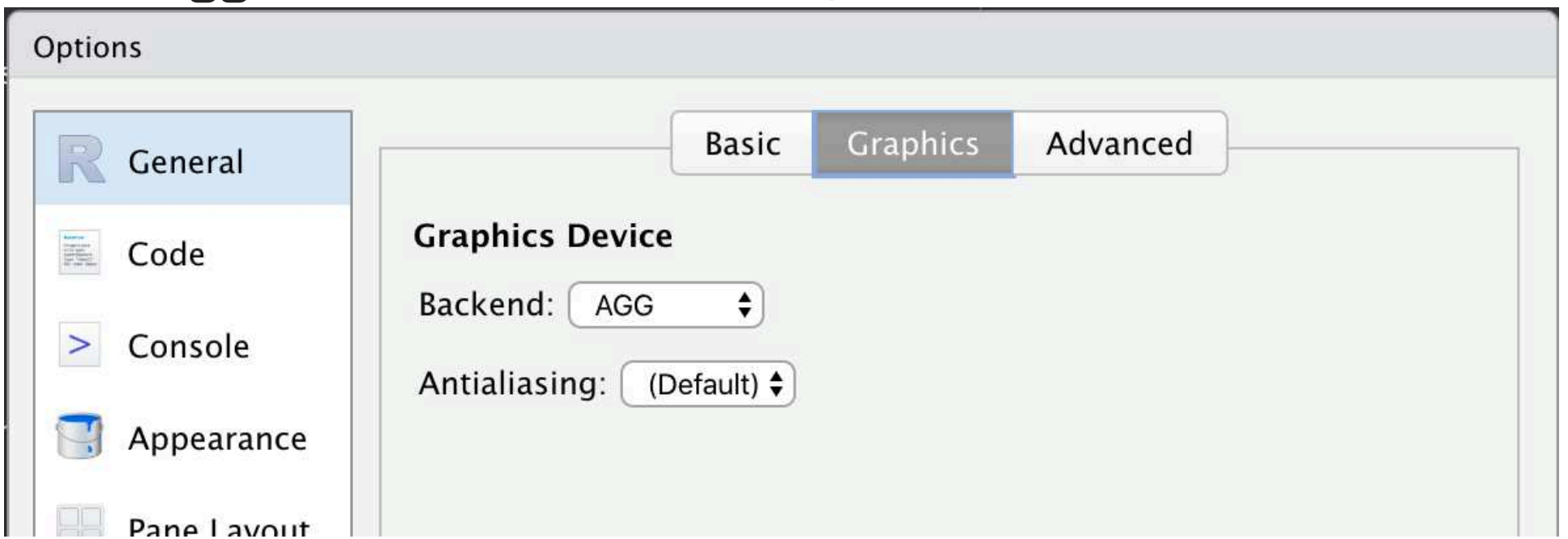
Source: tidyverse.org/blog/2021/02/modern-text-features

The {ragg} Package

Source: tidyverse.org/blog/2021/02/modern-text-features

The {ragg} Package

- use {ragg} when saving ggplots: `ggsave(device = ragg::agg_png)` (used by default if installed)
- use {ragg} in the *Rstudio Plots* pane:



- use {ragg} when knitting notebook files by setting `dev="ragg_png"` in the code chunk options.

Save the Graphic

```
ggsave("my_plot.png", width = 5, height = 6, dpi = 600, device = ragg::agg_png)
```

```
ggsave("my_plot.pdf", width = 5, height = 6, device = cairo_pdf)
```

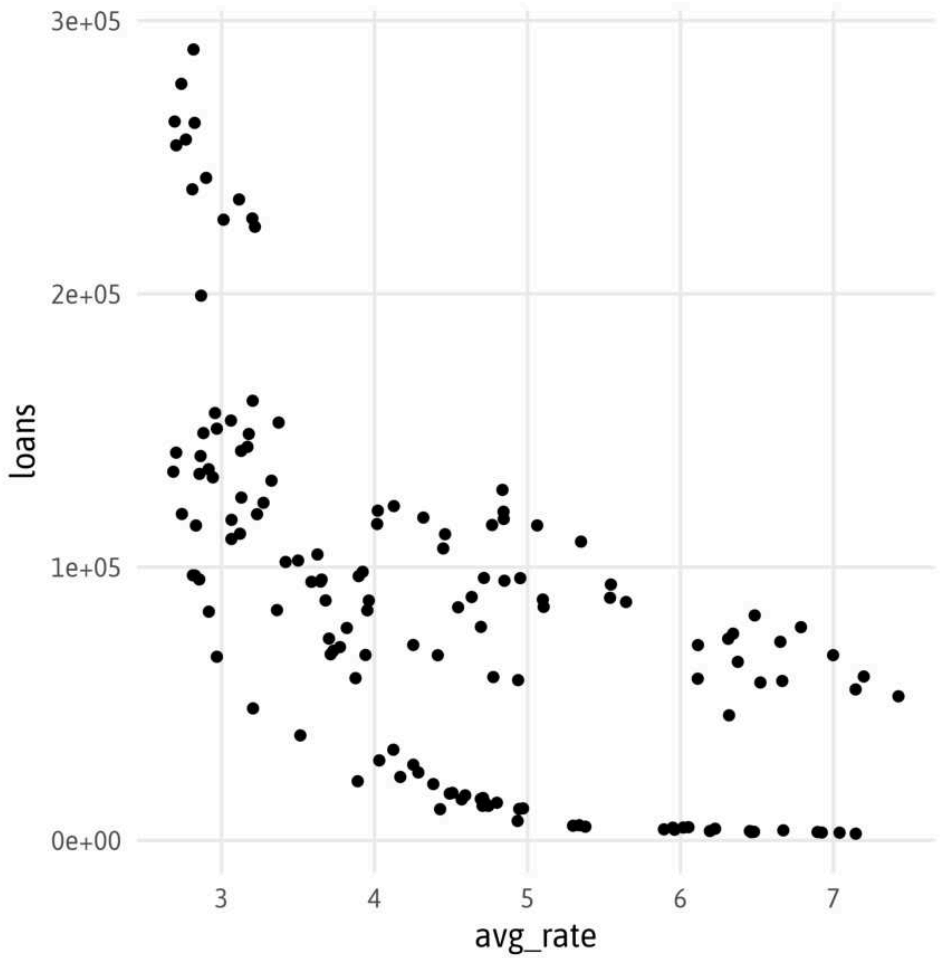
Statistical Layers

Cédric Scherer Data Visualization & Information Design

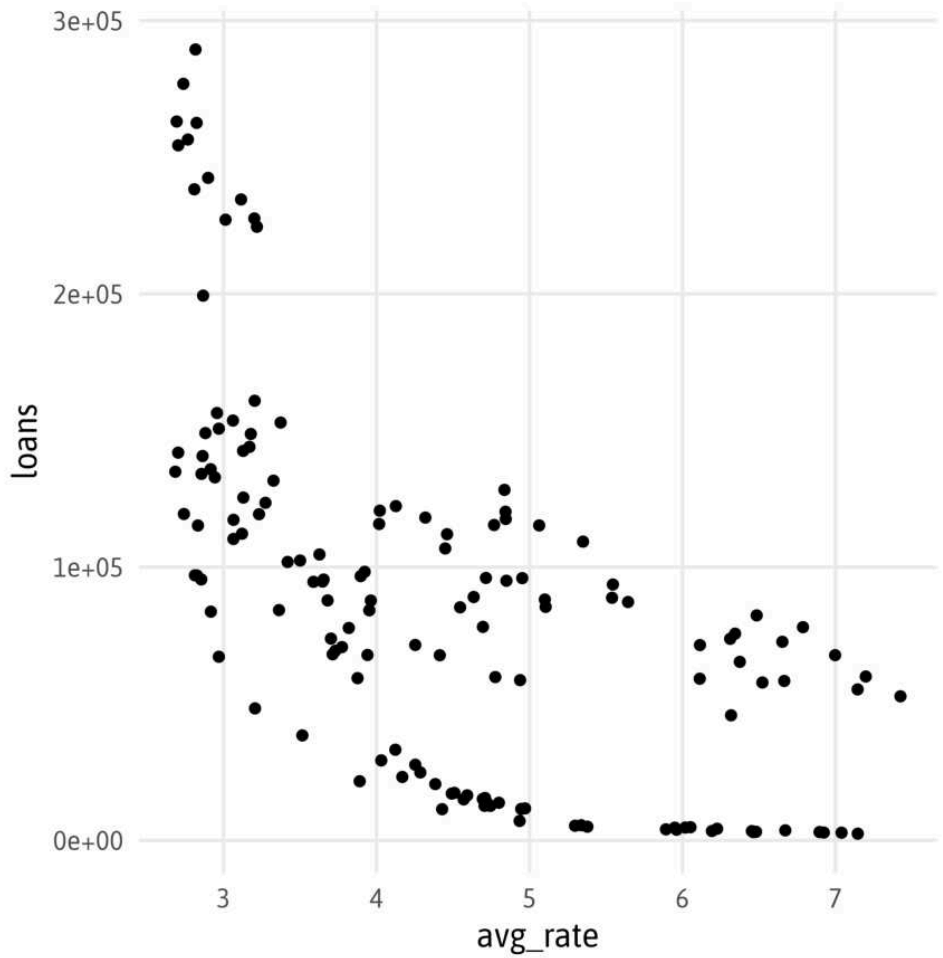
```
layer(data, mapping, geom, stat, position)
```

stat_* and geom_*

```
1 ggplot(fannie_agg, aes(x = avg_rate
2   geom_point(stat = "identity") # c
```

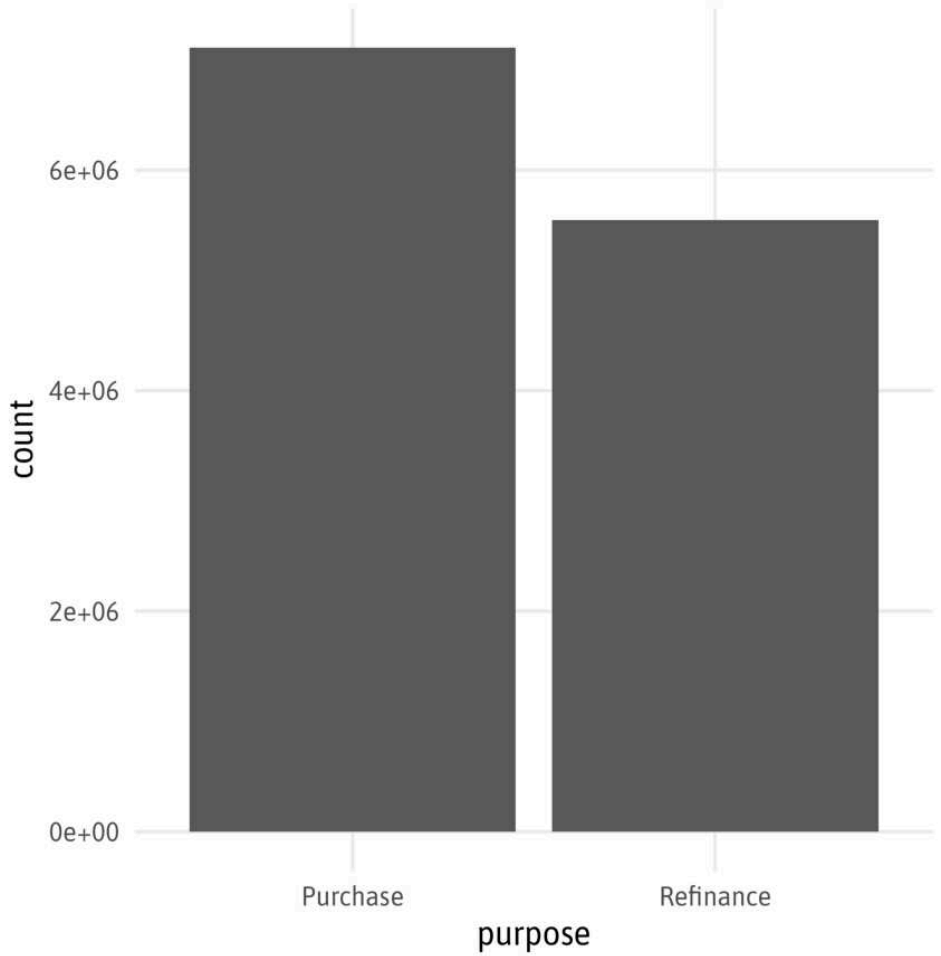



```
1 ggplot(fannie_agg, aes(x = avg_rate
2   stat_identity(geom = "point") # c
```

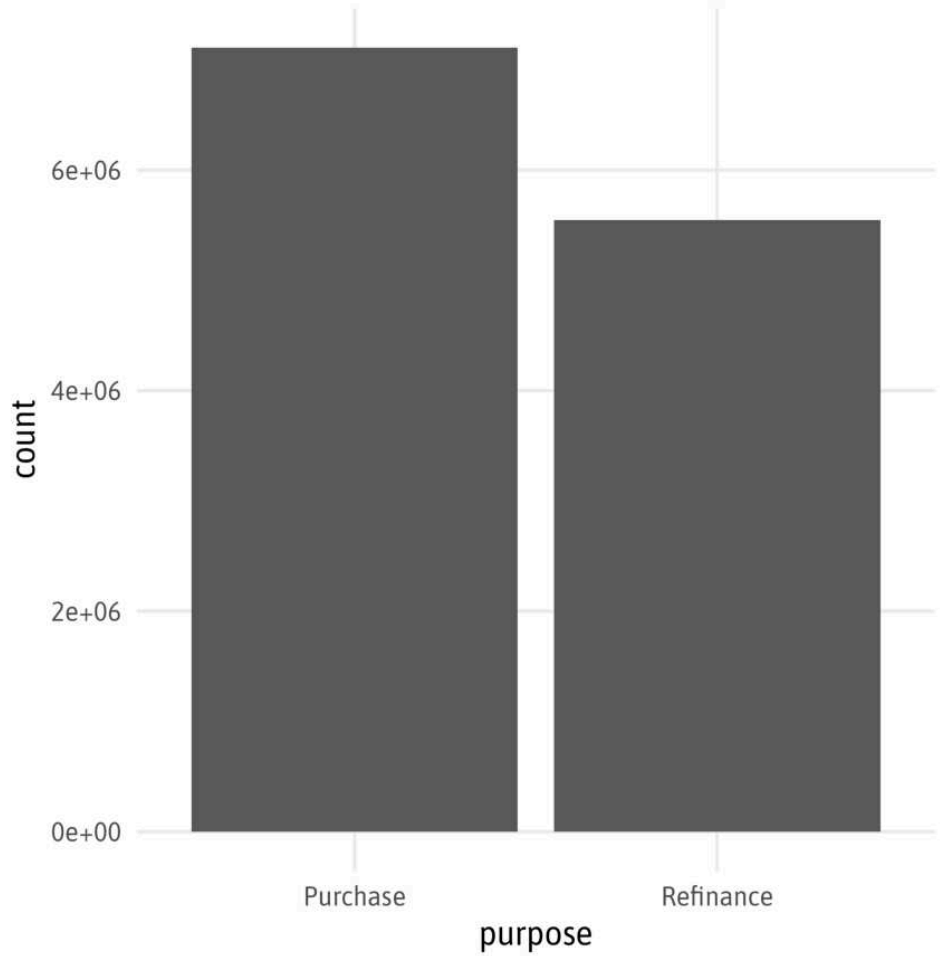


stat_* and geom_*

```
1 ggplot(fannie_raw, aes(x = purpose)
2   geom_bar(stat = "count") # defaul
```

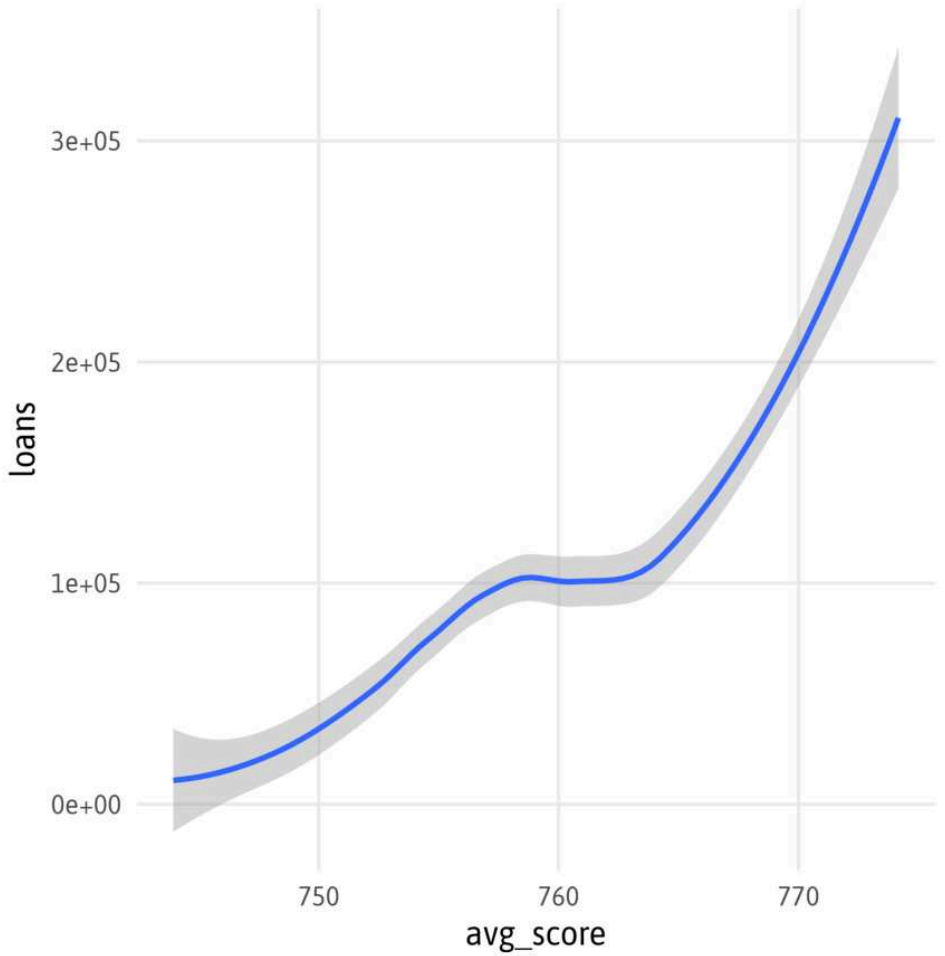



```
1 ggplot(fannie_raw, aes(x = purpose)
2   stat_count(geom = "bar") # defaul
```

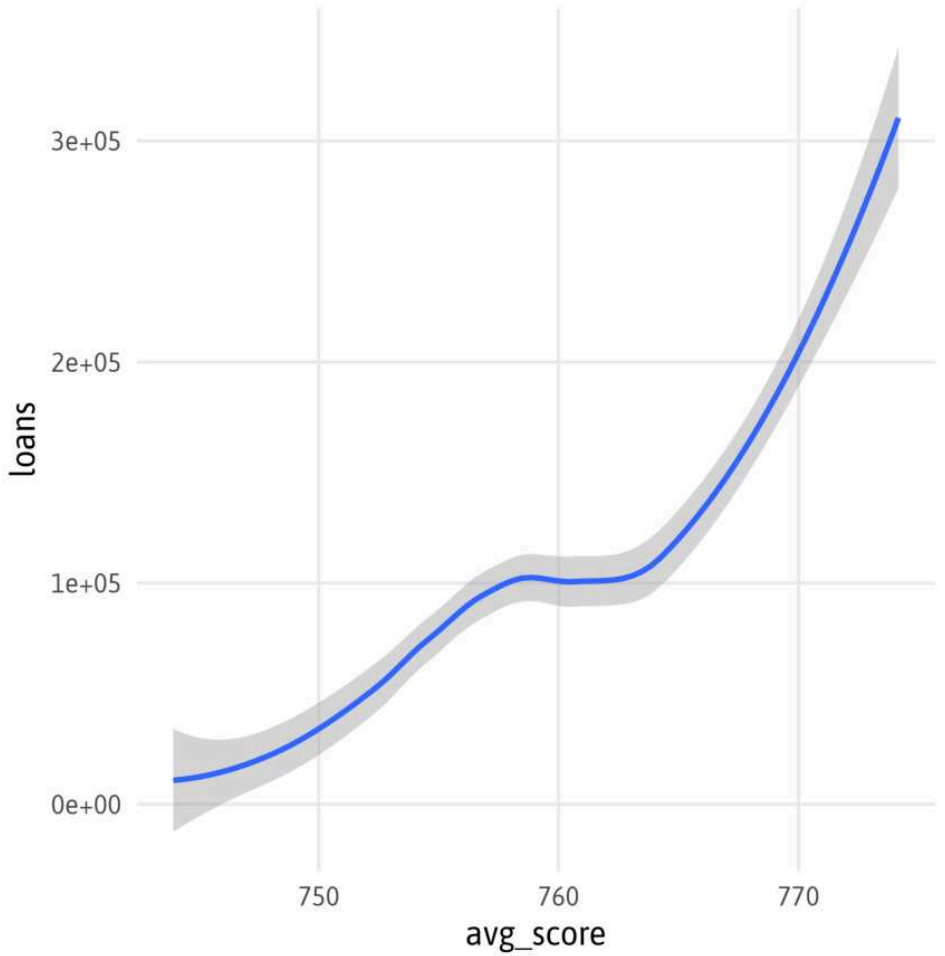


stat_* and geom_*

```
1 ggplot(fannie_agg, aes(x = avg_scor
2   geom_smooth(stat = "smooth") # de
```

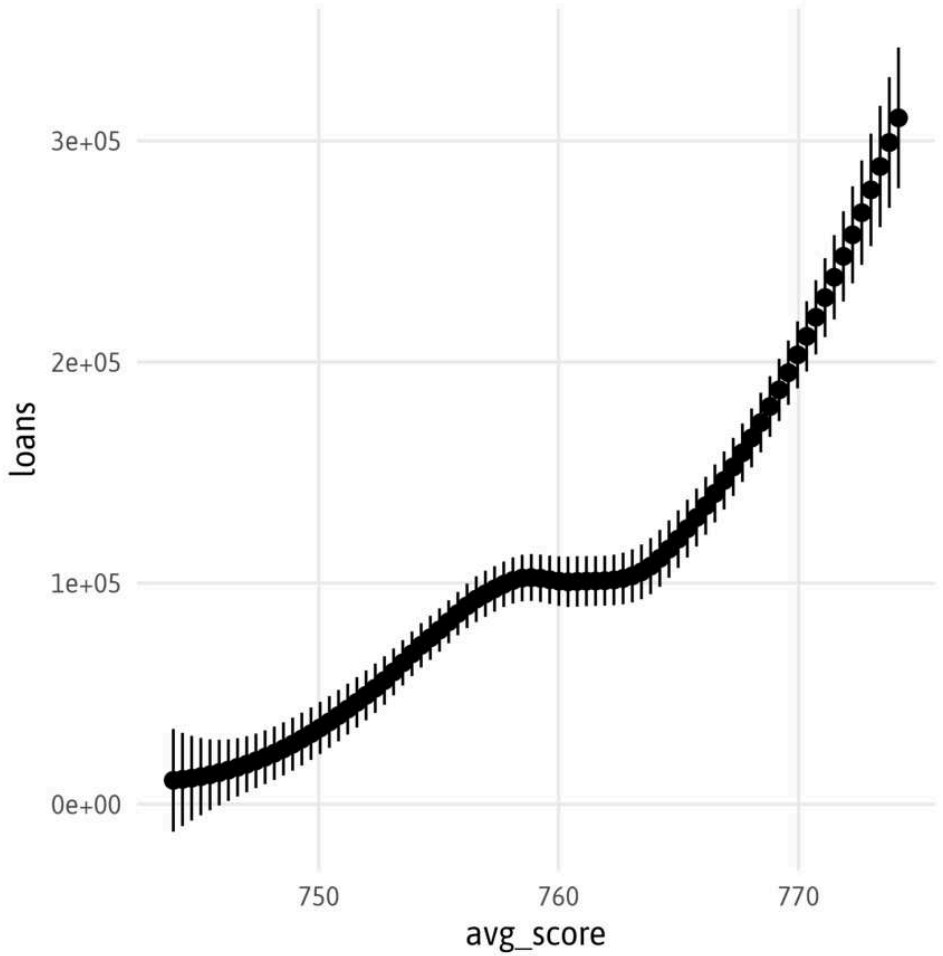



```
1 ggplot(fannie_agg, aes(x = avg_scor
2   stat_smooth(geom = "smooth") # de
```

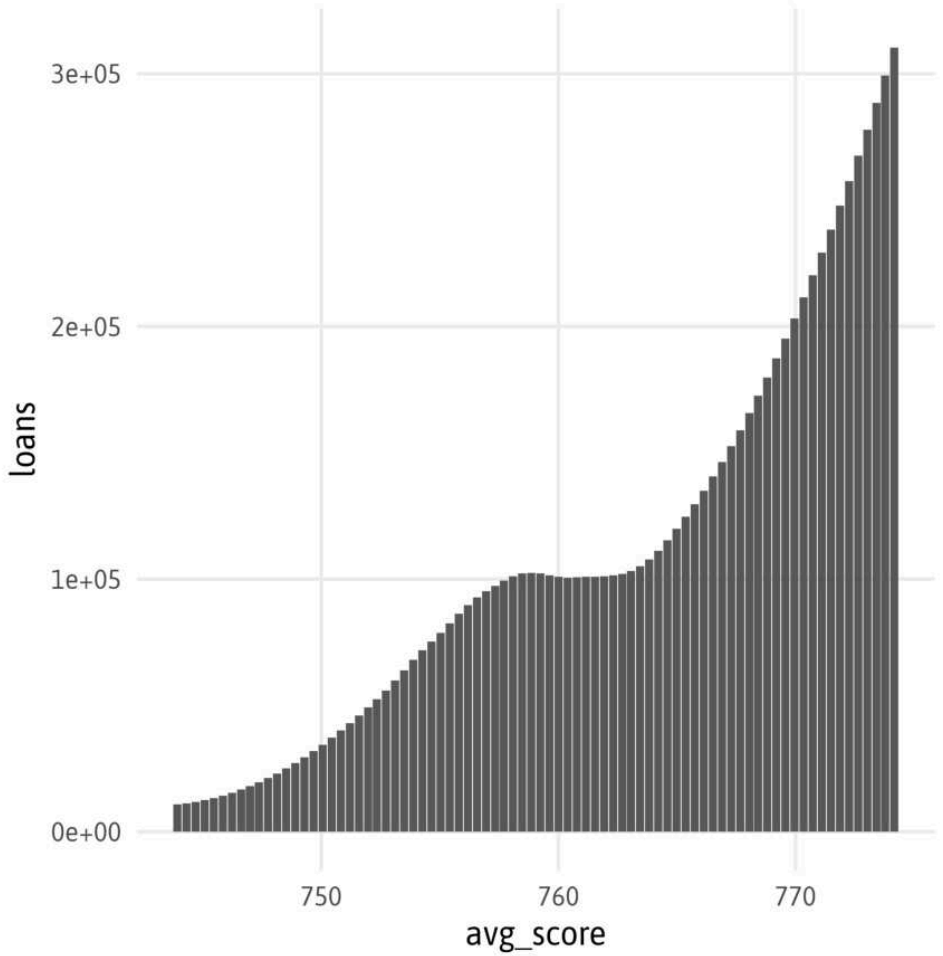


stat_* and geom_*

```
1 ggplot(fannie_agg, aes(x = avg_scor
2   stat_smooth(geom = "pointrange")
```

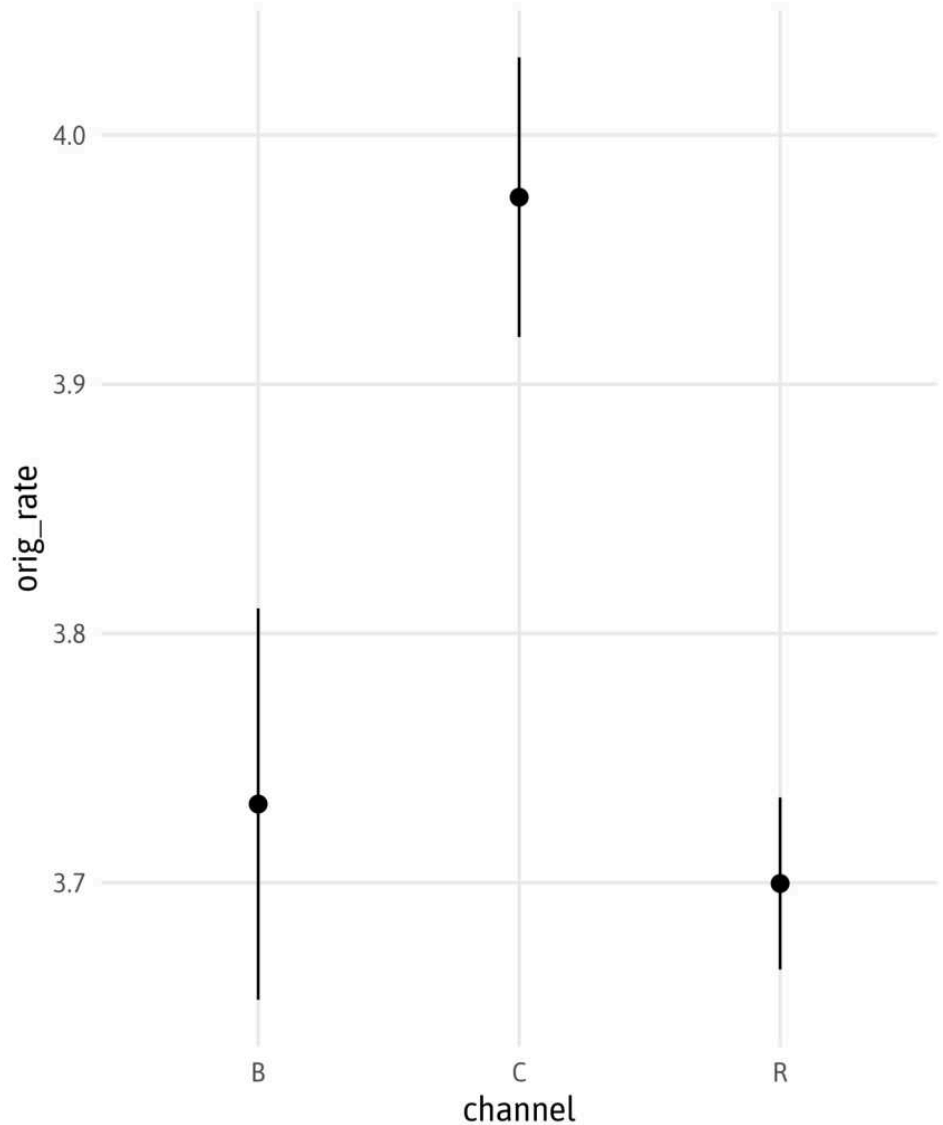



```
1 ggplot(fannie_agg, aes(x = avg_scor
2   stat_smooth(geom = "col")
```



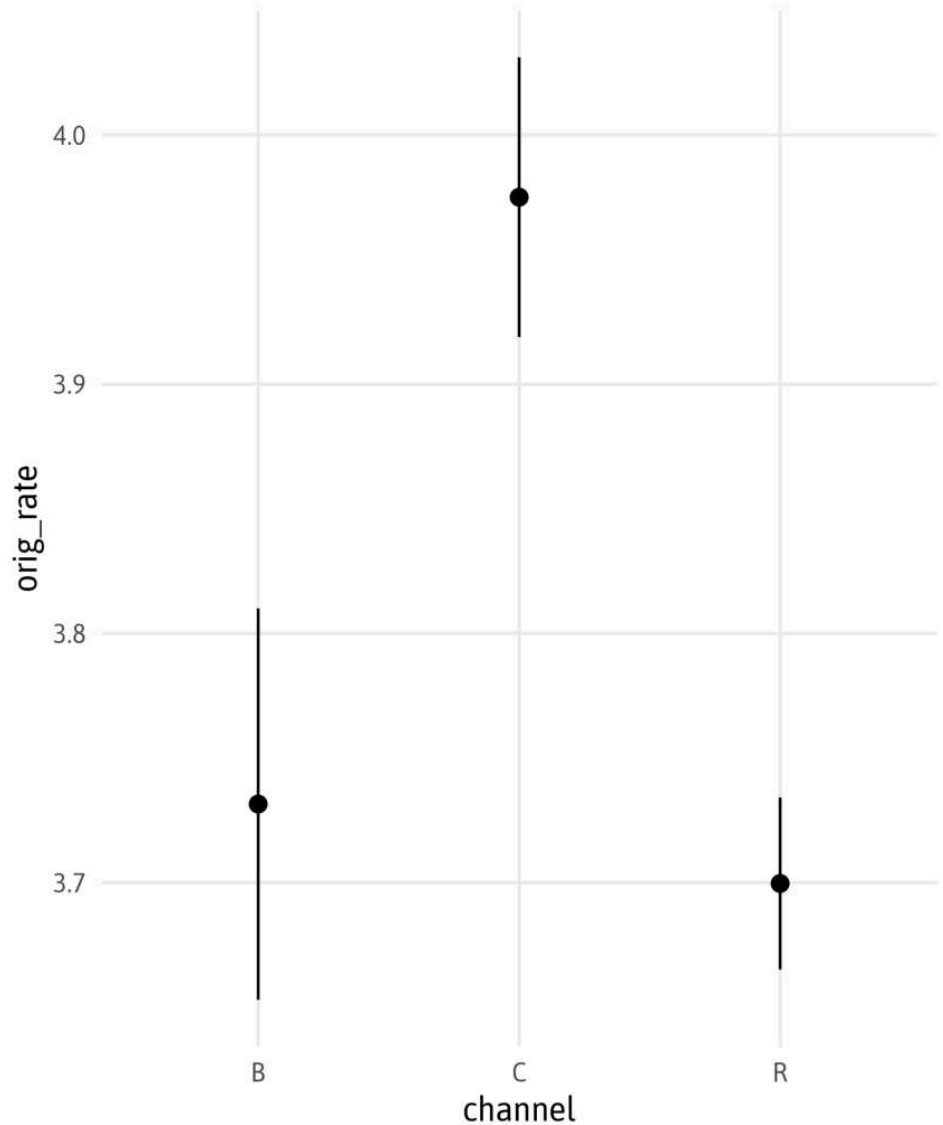
Statistical Summaries

```
1 ggplot(
2     fannie_sub,
3     aes(x = channel, y = orig_rate)
4   ) +
5   stat_summary()
```



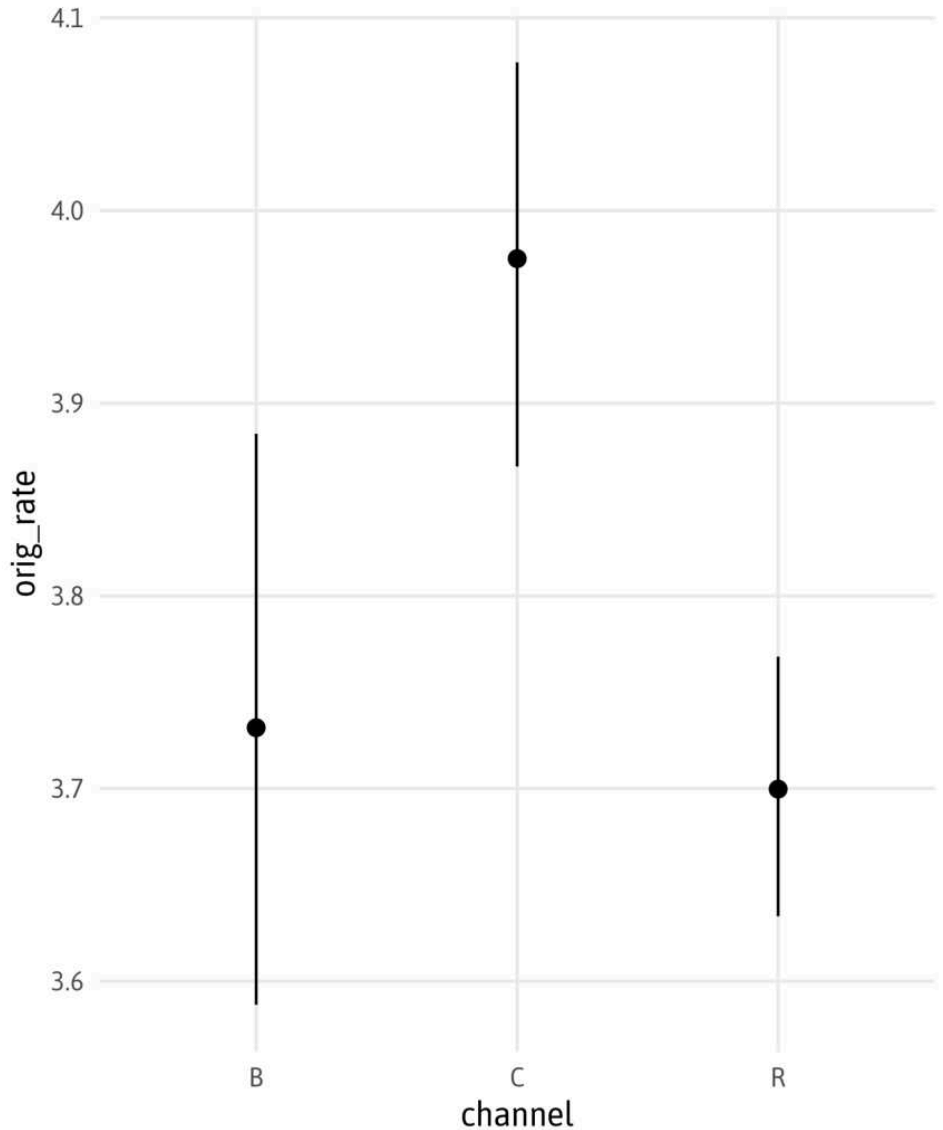
Statistical Summaries

```
ggplot(
    fannie_sub,
    aes(x = channel, y = orig_rate)
  ) +
  stat_summary(
    fun.data = "mean_se", # default
    geom = "pointrange" # default
  )
```



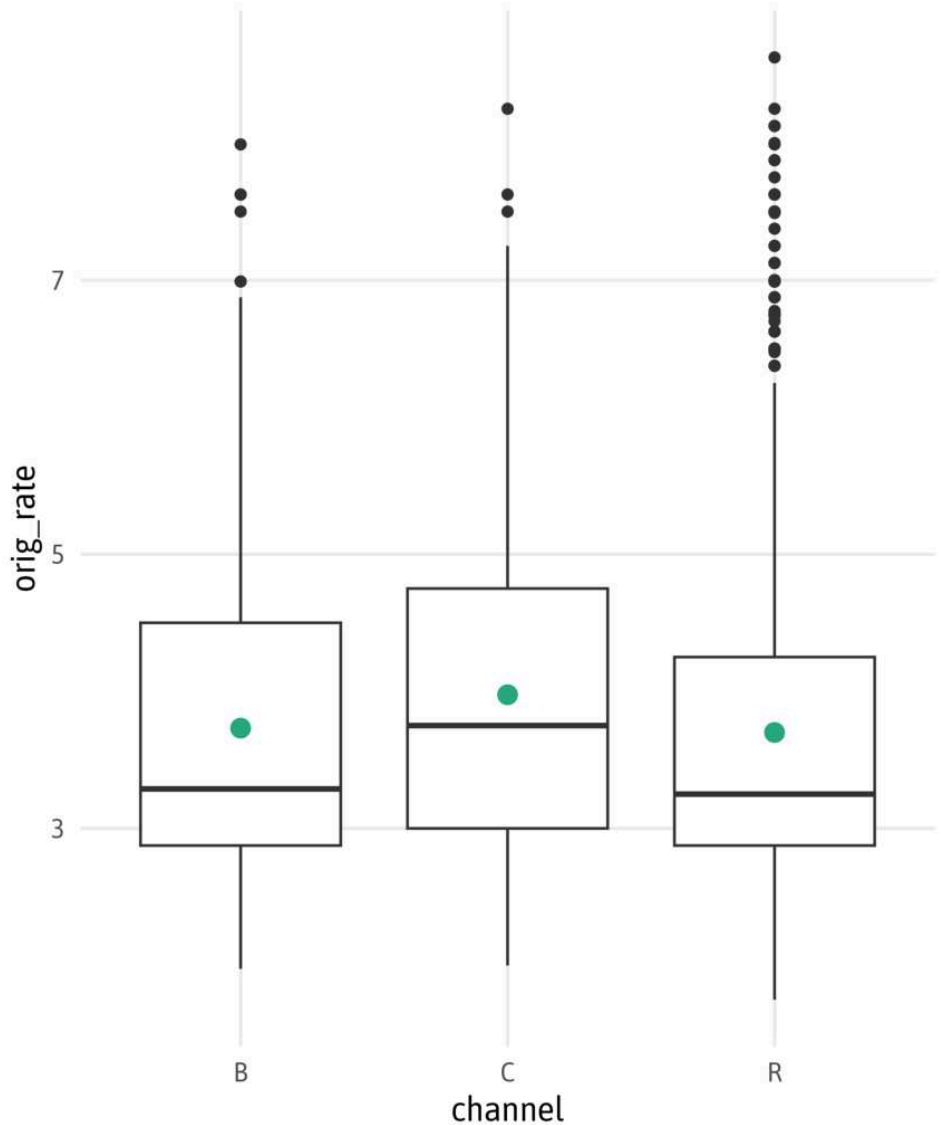
Statistical Summaries

```
ggplot(
    fannie_sub,
    aes(x = channel, y = orig_rate)
  ) +
  stat_summary(
    fun.data = "mean_cl_boot",
    geom = "pointrange" # default
  )
```



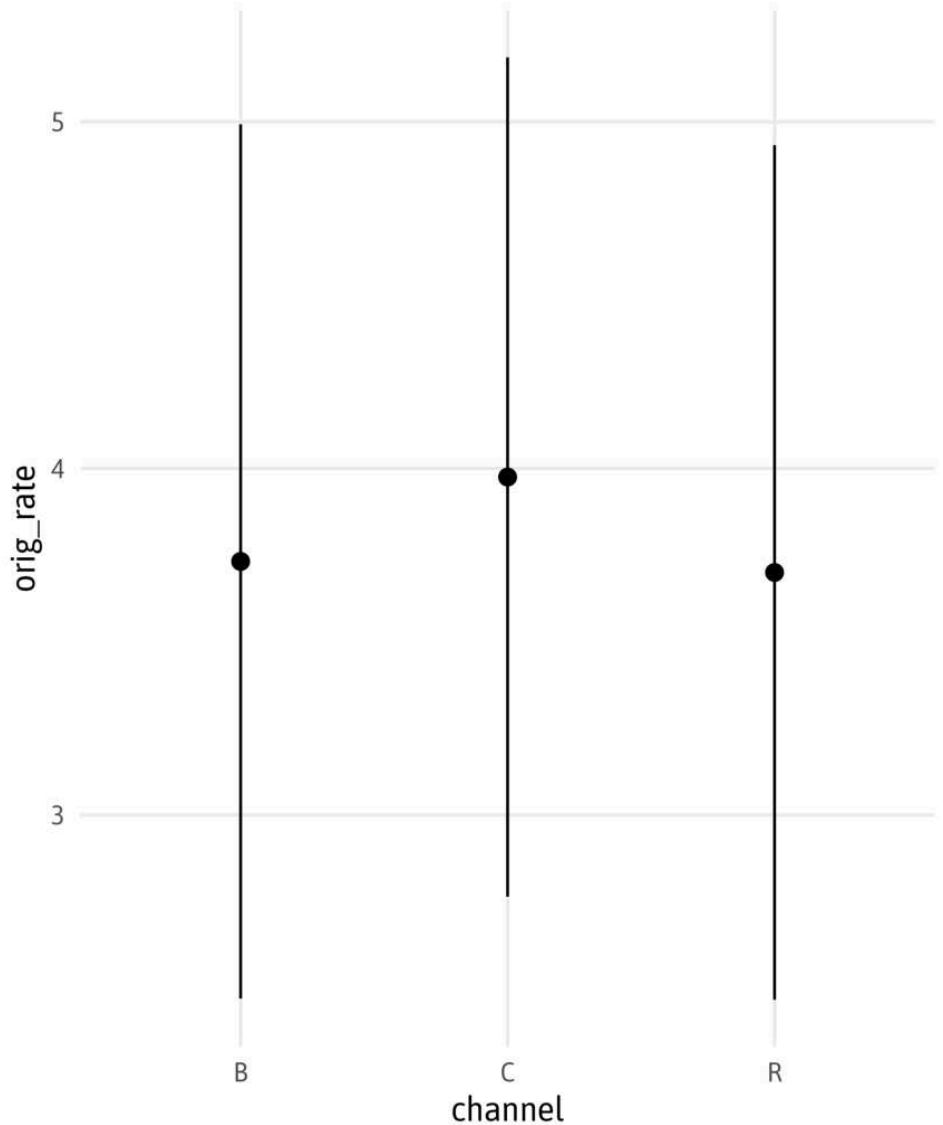
Statistical Summaries

```
ggplot(
    fannie_sub,
    aes(x = channel, y = orig_rate)
  ) +
  geom_boxplot() +
  stat_summary(
    fun = "mean",
    geom = "point",
    color = "#28a87d",
    size = 3
  )
```



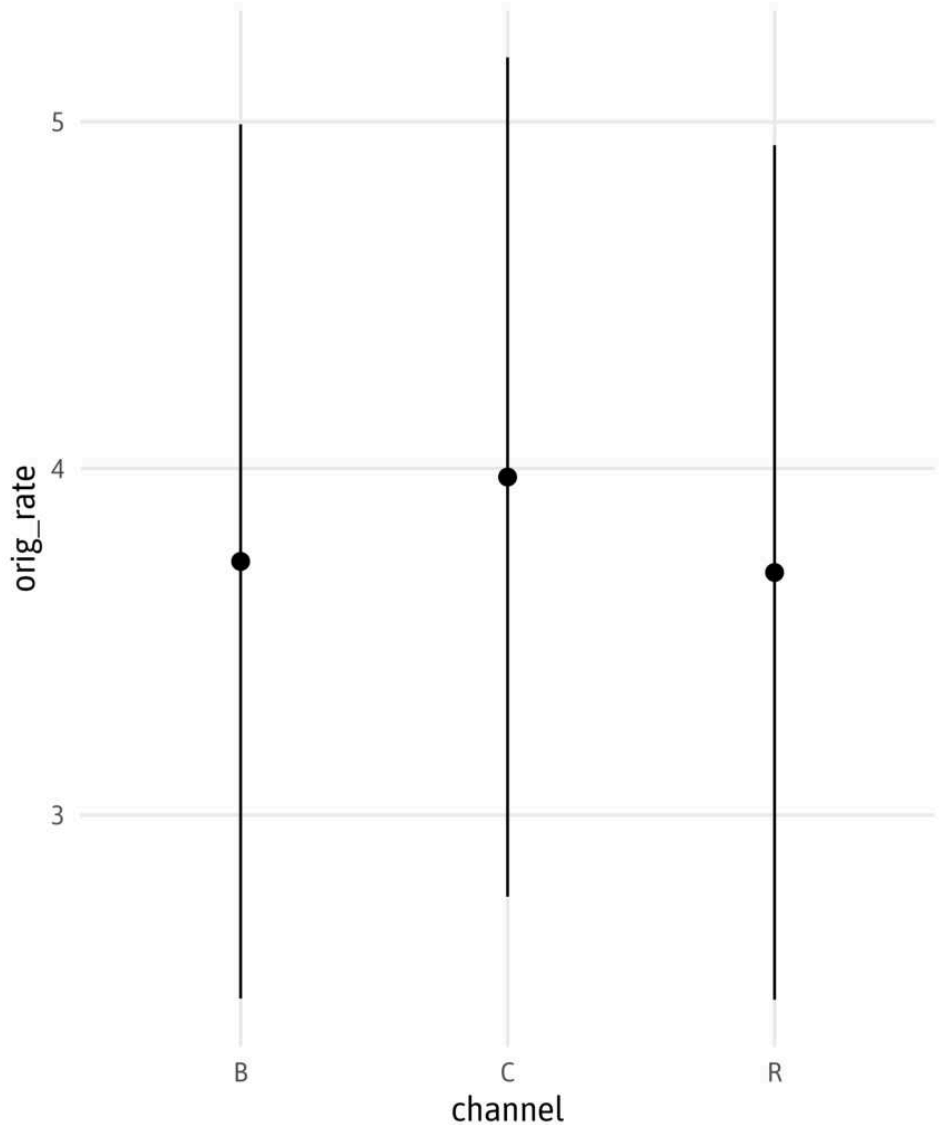
Statistical Summaries

```
ggplot(
    fannie_sub,
    aes(x = channel, y = orig_rate)
  ) +
  stat_summary(
    fun = "mean",
    fun.max = function(y) mean(y) +
    fun.min = function(y) mean(y) -
  )
```



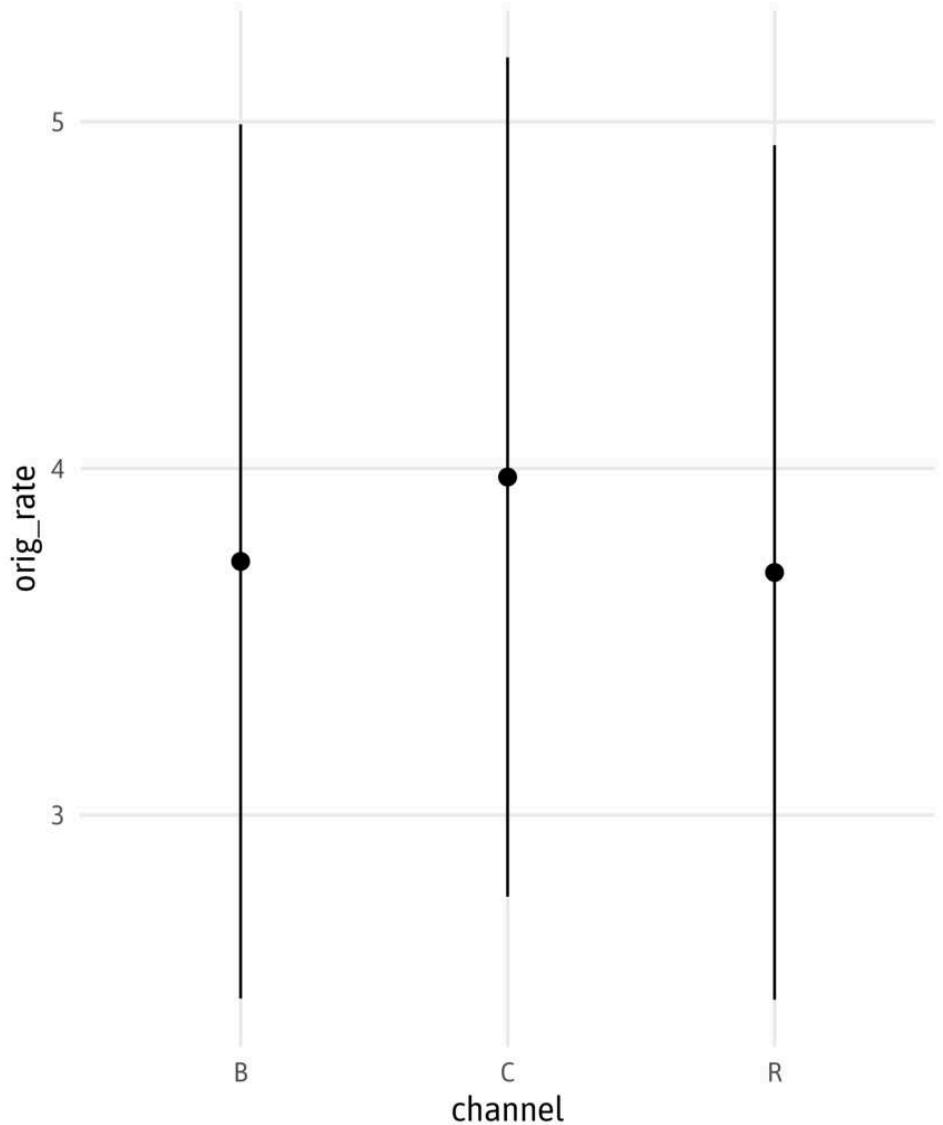
Statistical Summaries

```
1 ggplot(
2     fannie_sub,
3     aes(x = channel, y = orig_rate)
4   ) +
5   stat_summary(
6     fun = "mean",
7     fun.max = function(foo) mean(fo
8     fun.min = function(foo) mean(fo
9   )
```



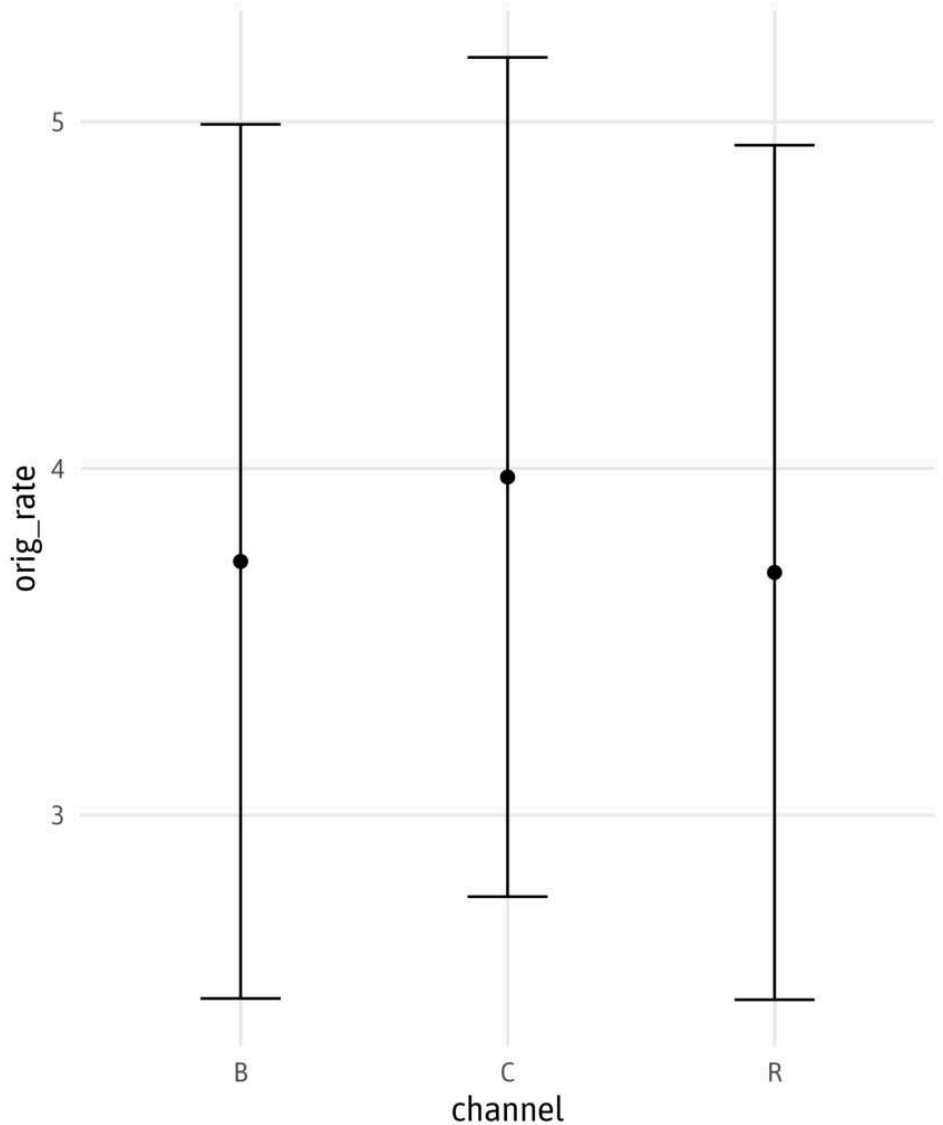
Statistical Summaries

```
ggplot(
    fannie_sub,
    aes(x = channel, y = orig_rate)
  ) +
  stat_summary(
    fun.data = "mean_sdl",
    fun.args = list(mult = 1)
  )
```



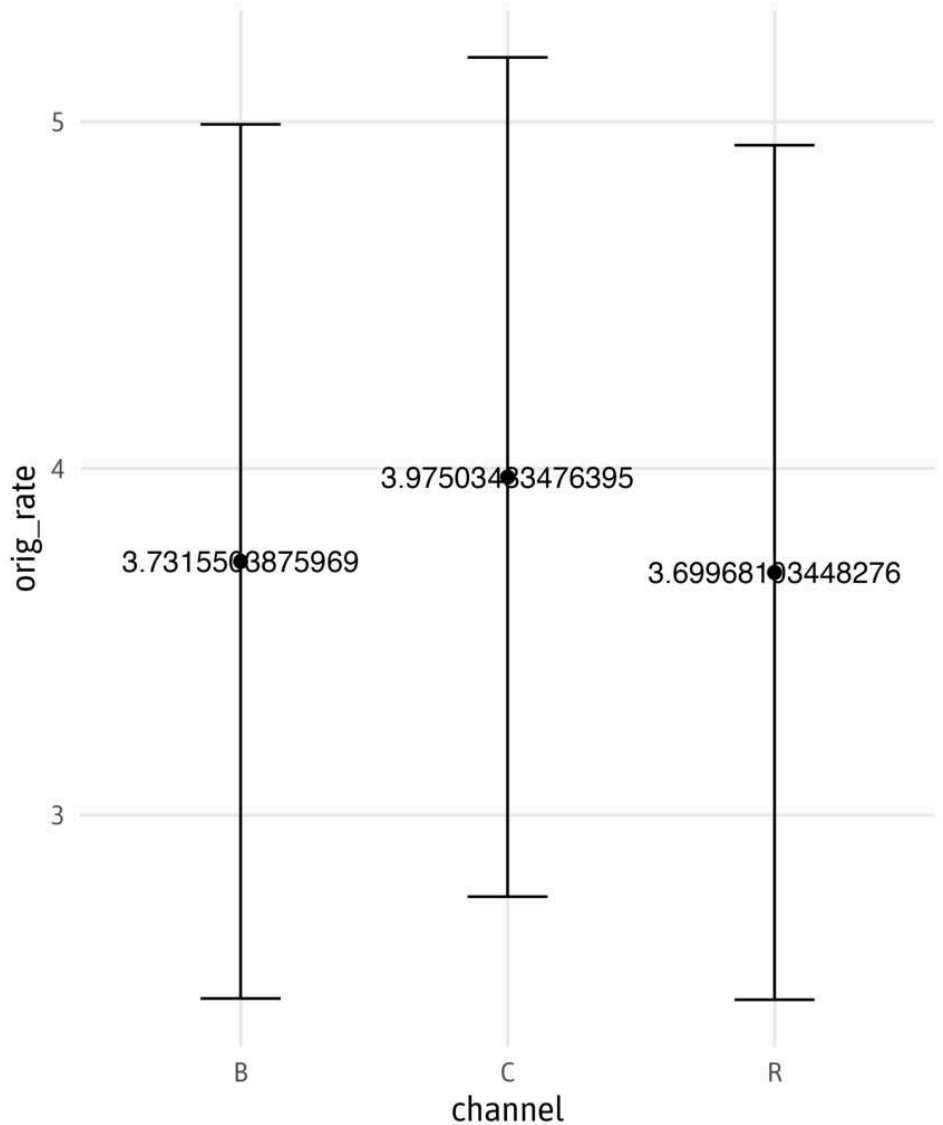
Statistical Summaries

```
ggplot(
    fannie_sub,
    aes(x = channel, y = orig_rate)
  ) +
  stat_summary(
    geom = "errorbar",
    fun.max = function(y) mean(y) +
    fun.min = function(y) mean(y) -
    width = .3
  ) +
  stat_summary(
    geom = "point",
    fun = "mean",
    size = 2
  )
```



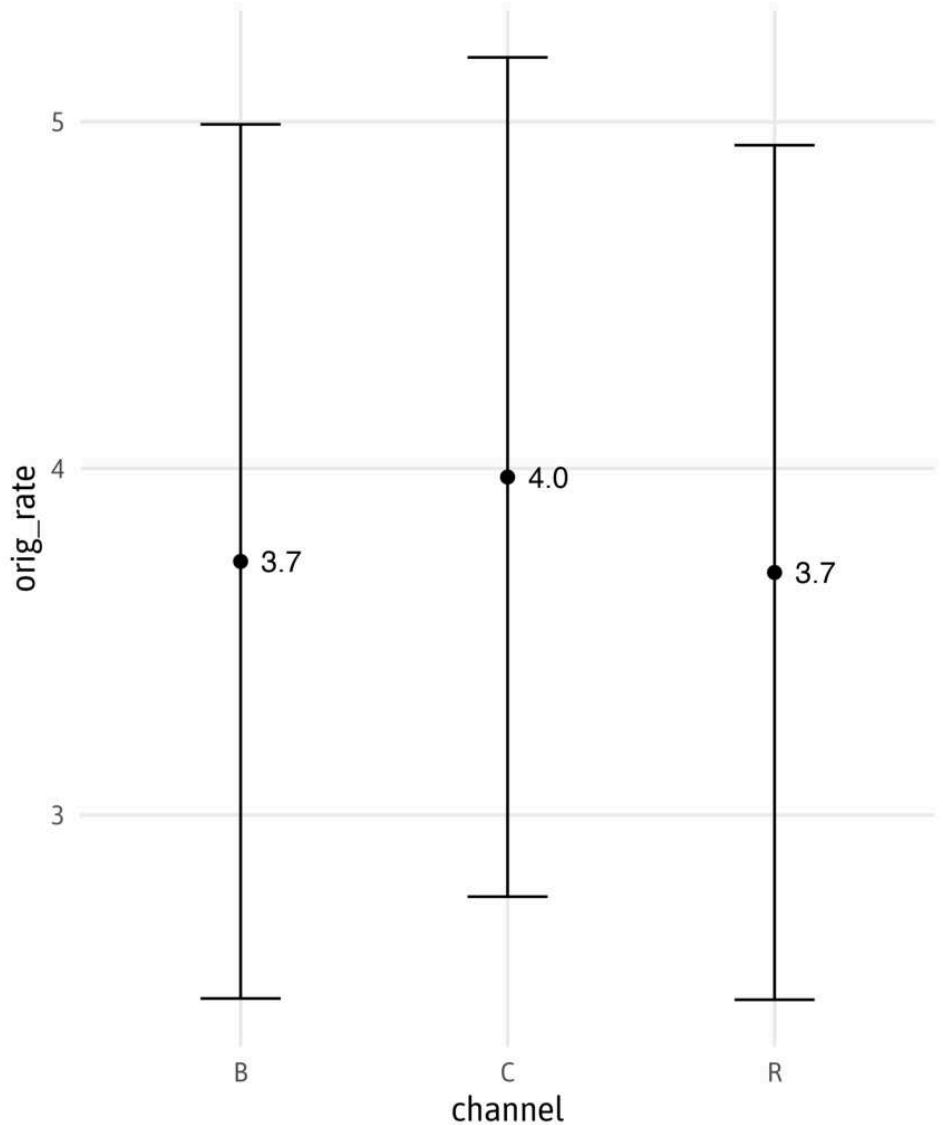
Statistical Summaries

```
ggplot(
    fannie_sub,
    aes(x = channel, y = orig_rate)
  ) +
  stat_summary(
    geom = "errorbar",
    fun.max = function(y) mean(y) +
    fun.min = function(y) mean(y) -
    width = .3
  ) +
  stat_summary(
    geom = "point",
    fun = "mean",
    size = 2
  ) +
  stat_summary(
    geom = "text",
    fun = "mean",
    aes(label = after_stat(y))
  )
```



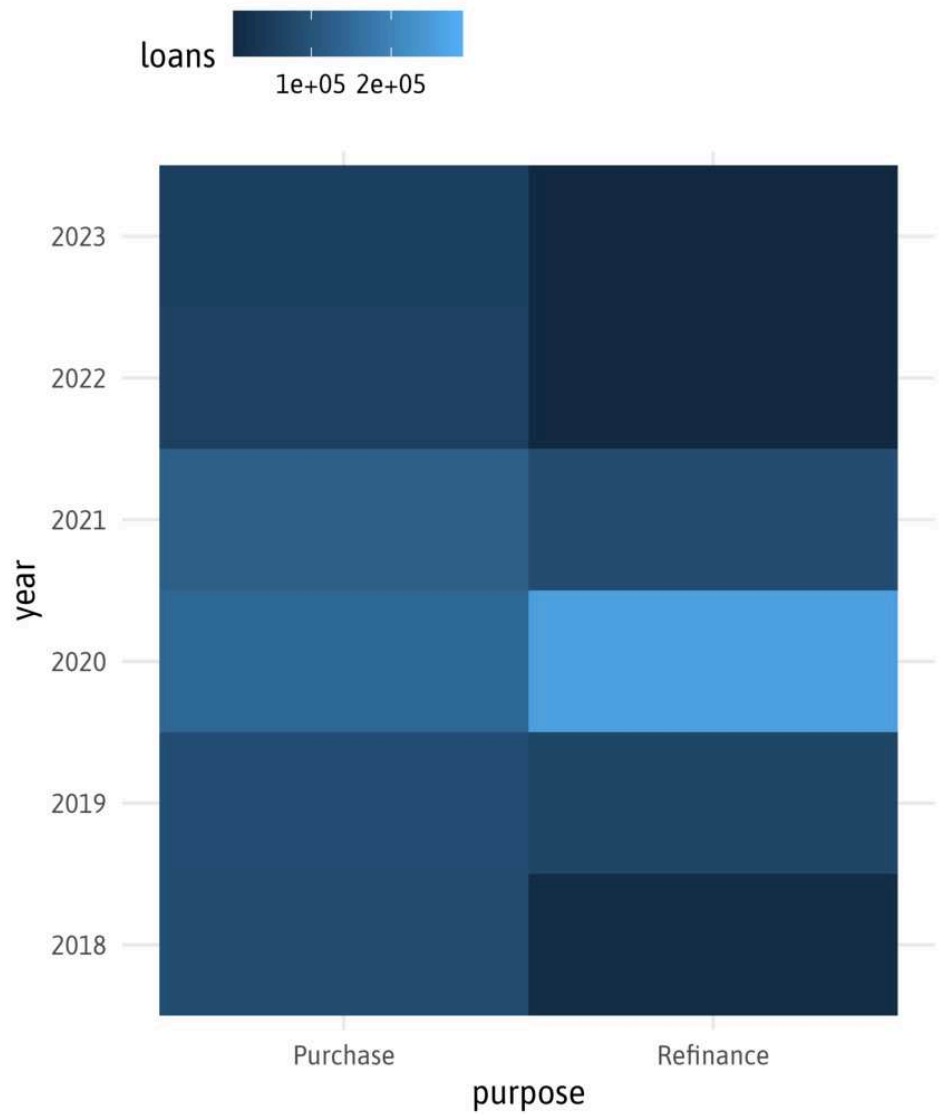
Statistical Summaries

```
ggplot(
    fannie_sub,
    aes(x = channel, y = orig_rate)
  ) +
  stat_summary(
    geom = "errorbar",
    fun.max = function(y) mean(y) +
    fun.min = function(y) mean(y) -
    width = .3
  ) +
  stat_summary(
    geom = "point",
    fun = "mean",
    size = 2
  ) +
  stat_summary(
    geom = "text",
    fun = "mean",
    aes(label = after_stat(
      sprintf("%2.1f", y)
    )),
    hjust = -.5
  )
```



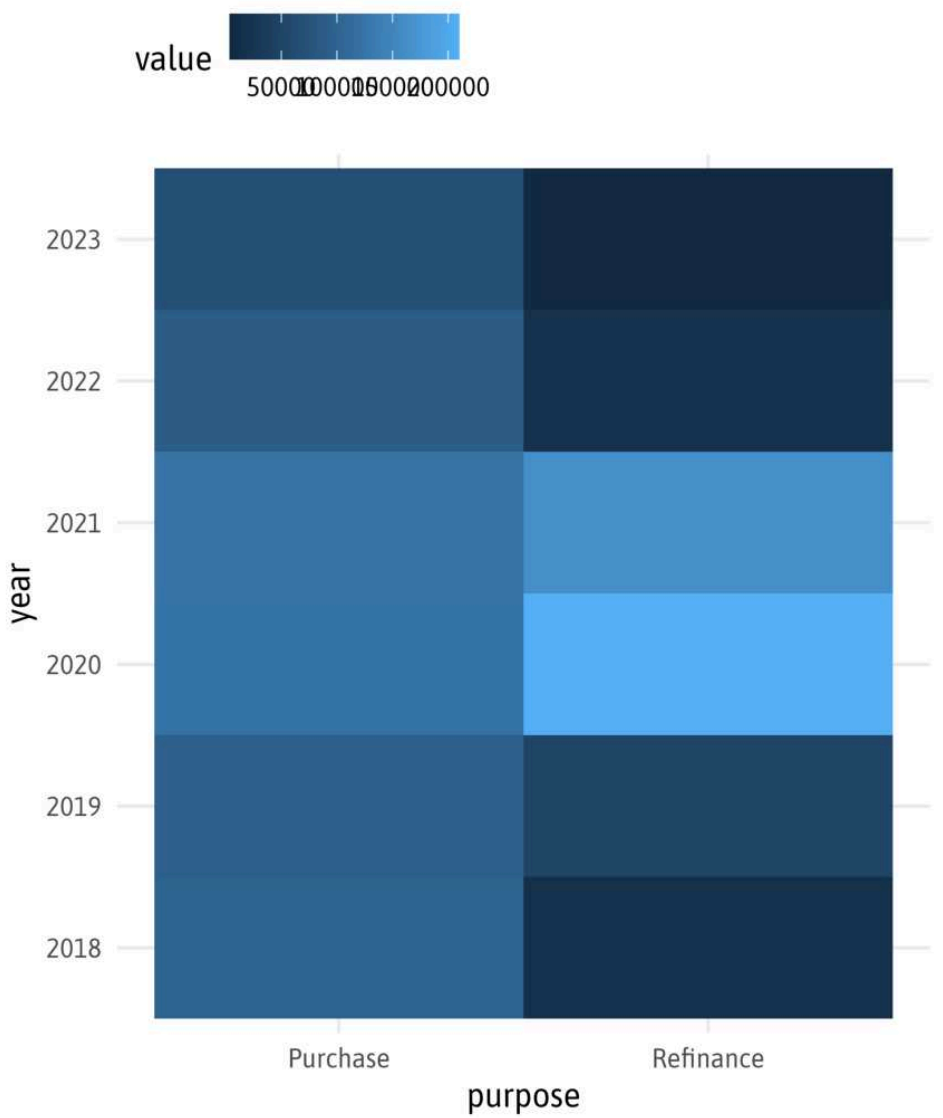
Summarize First!

```
1  ggplot(
2      fannie_agg,
3      aes(x = purpose,
4          y = year,
5          fill = loans)
6    ) +
7    geom_tile()
```



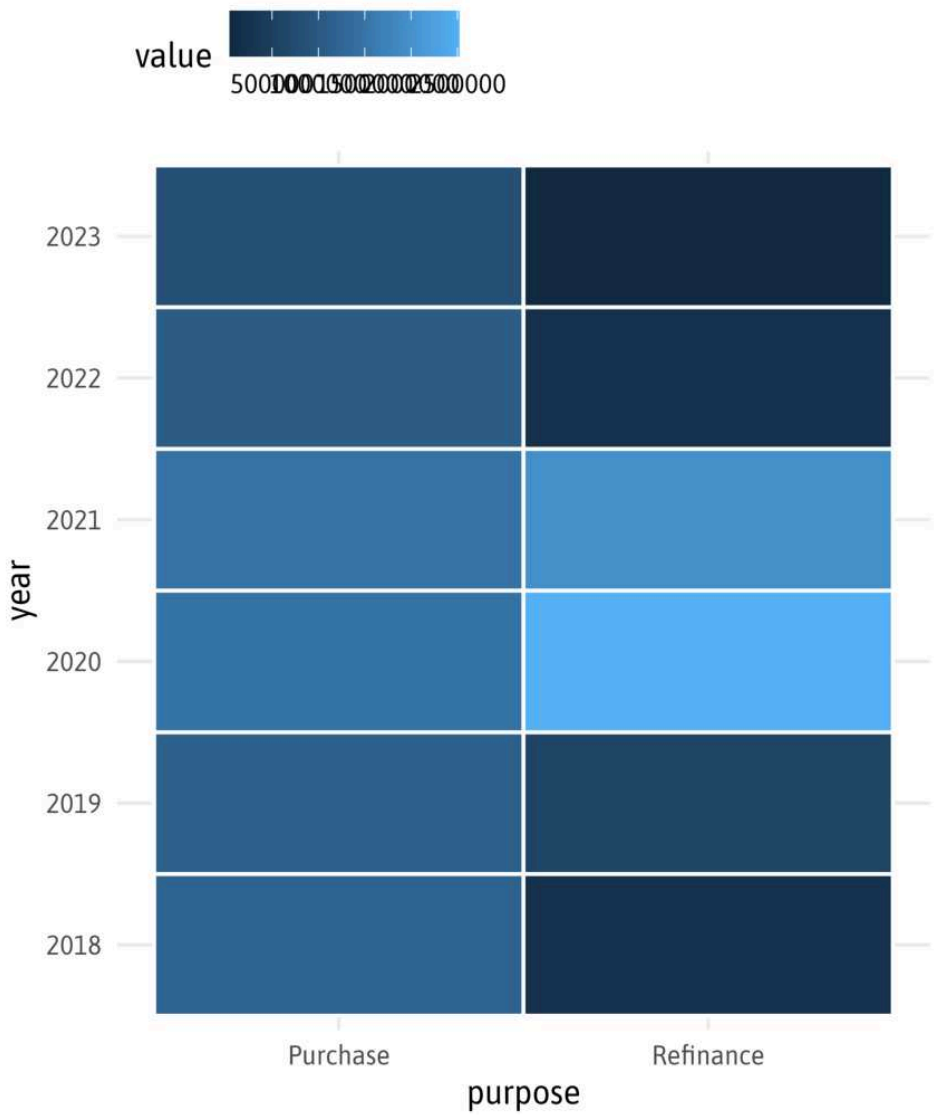
Statistical Summaries

```
1 ggplot(
2     fannie_agg,
3     aes(x = purpose,
4         y = year,
5         z = loans)
6   ) +
7   stat_summary_2d()
```



Statistical Summaries

```
ggplot(
    fannie_agg,
    aes(x = purpose,
        y = year,
        z = loans)
) +
stat_summary_2d(
    geom = "tile",
    fun = "sum",
    color = "white",
    linewidth = .7
)
```

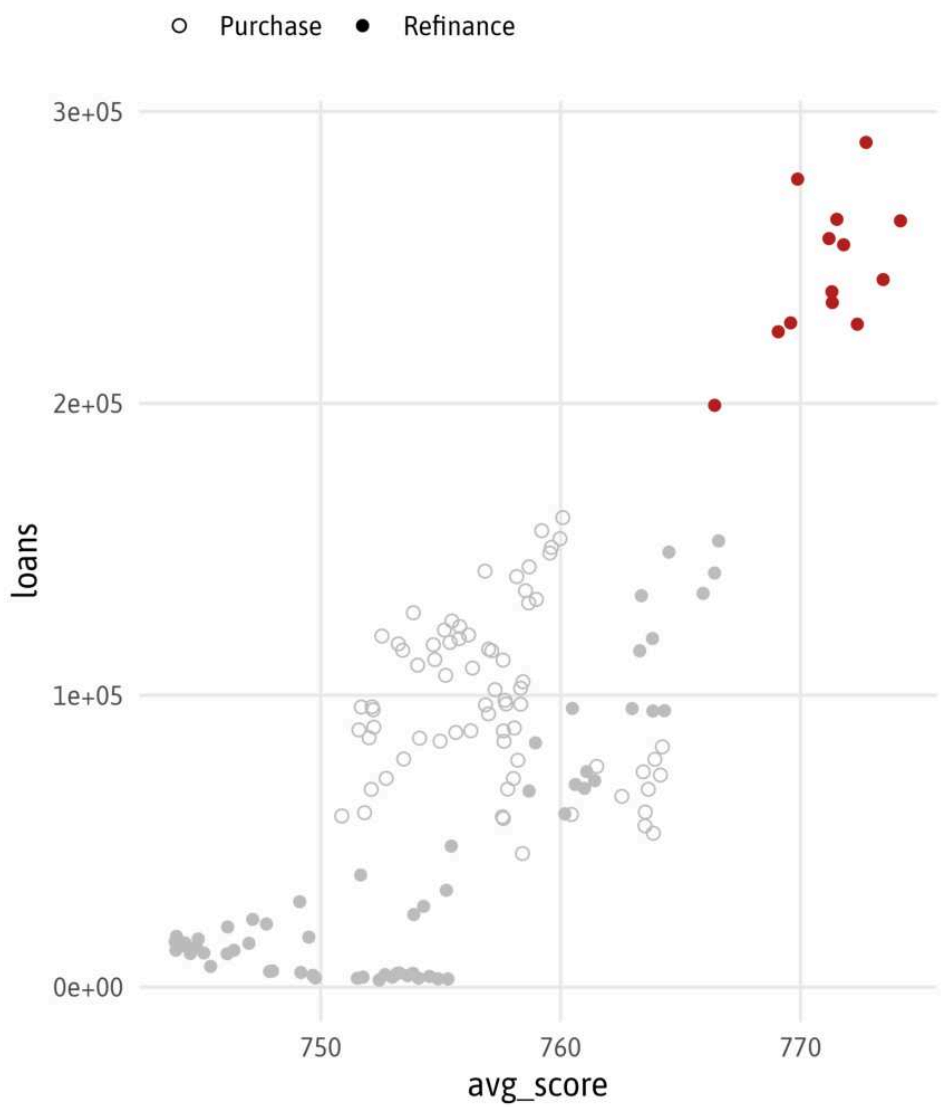


Annotations & Call-Outs

Cédric Scherer Data Visualization & Information Design

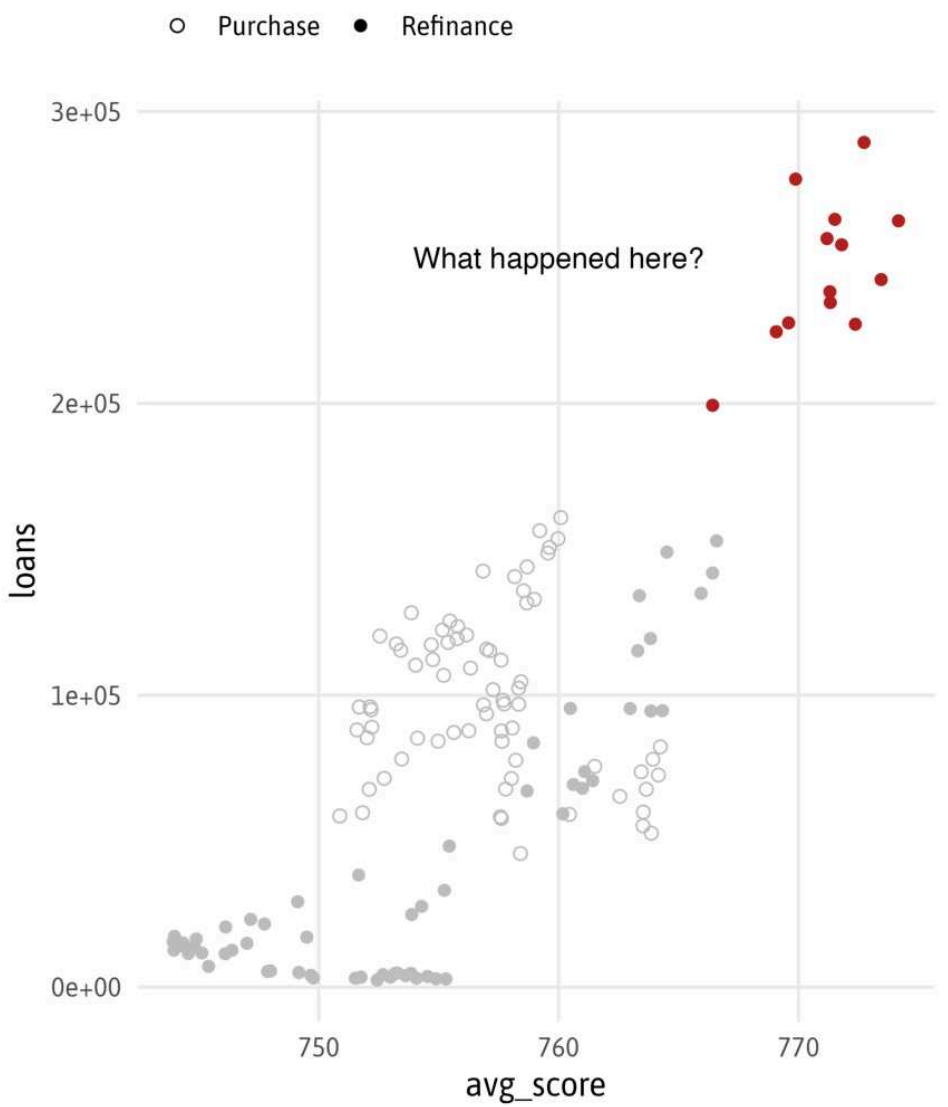
Add Single Text Annotations

```
ga <-
  ggplot(fannie_agg,
         aes(x = avg_score, y = loa
  geom_point(
    aes(color = loans > 190000,
        shape = purpose),
    size = 2
  ) +
  scale_color_manual(
    values = c("grey", "firebrick")
    guide = "none"
  ) +
  scale_shape_manual(
    values = c(21, 16),
    name = NULL
  ) +
  coord_cartesian(
    clip = "off"
  )

ga
```



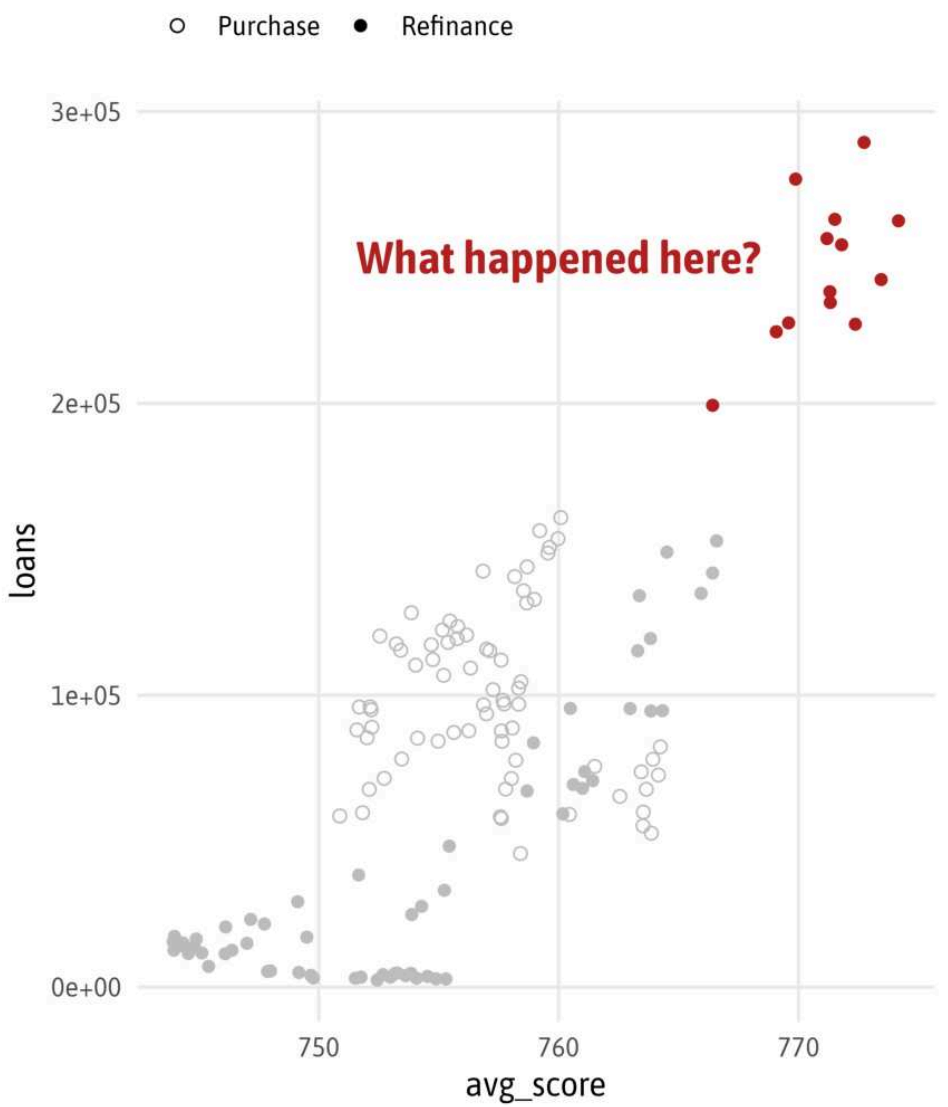
Add Single Text Annotations

```
1 ga +
2   annotate(
3     geom = "text",
4     x = 760,
5     y = 250000,
6     label = "What happened here?"
7   )
```



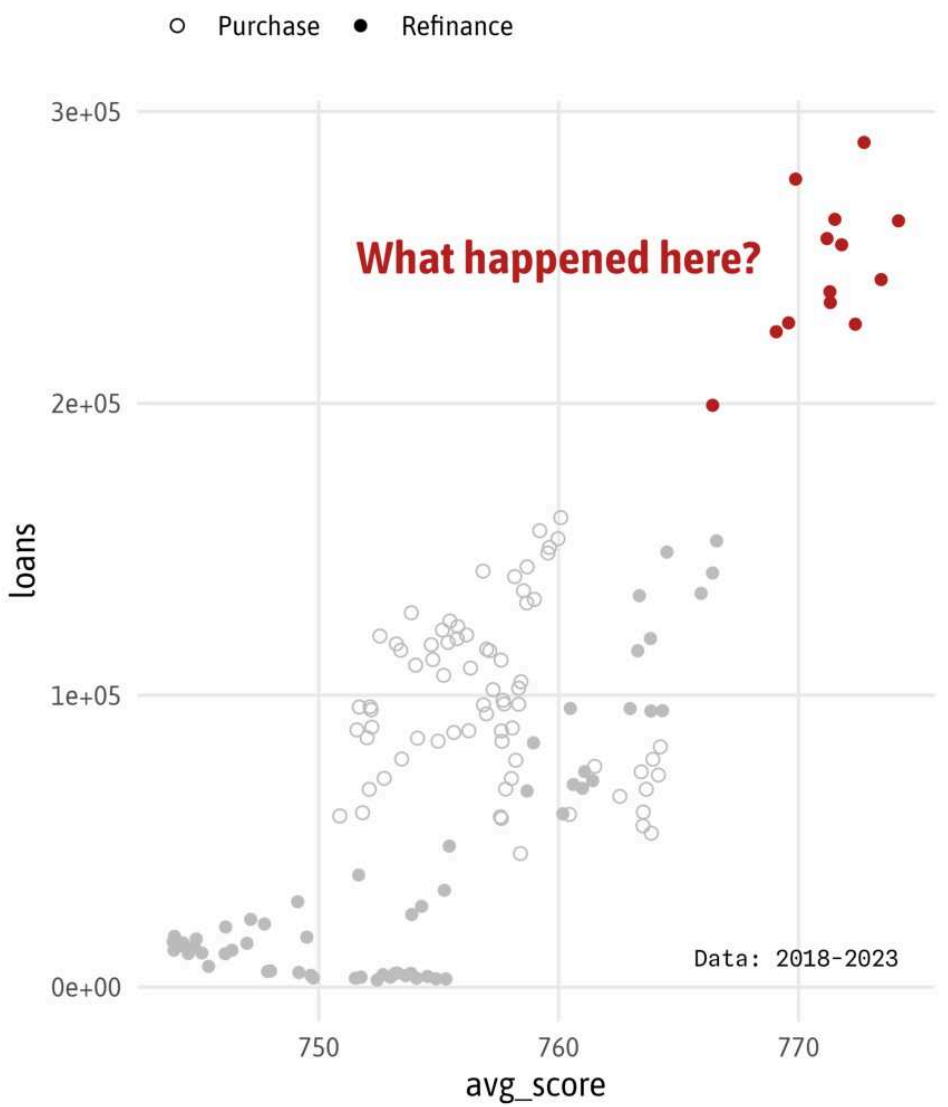
Style Text Annotations

```
ga +
  annotate(
    geom = "text",
    x = 760,
    y = 250000,
    label = "What happened here?",
    color = "firebrick",
    size = 6,
    family = "Asap SemiCondensed",
    fontface = "bold",
    lineheight =  .8
  )
```



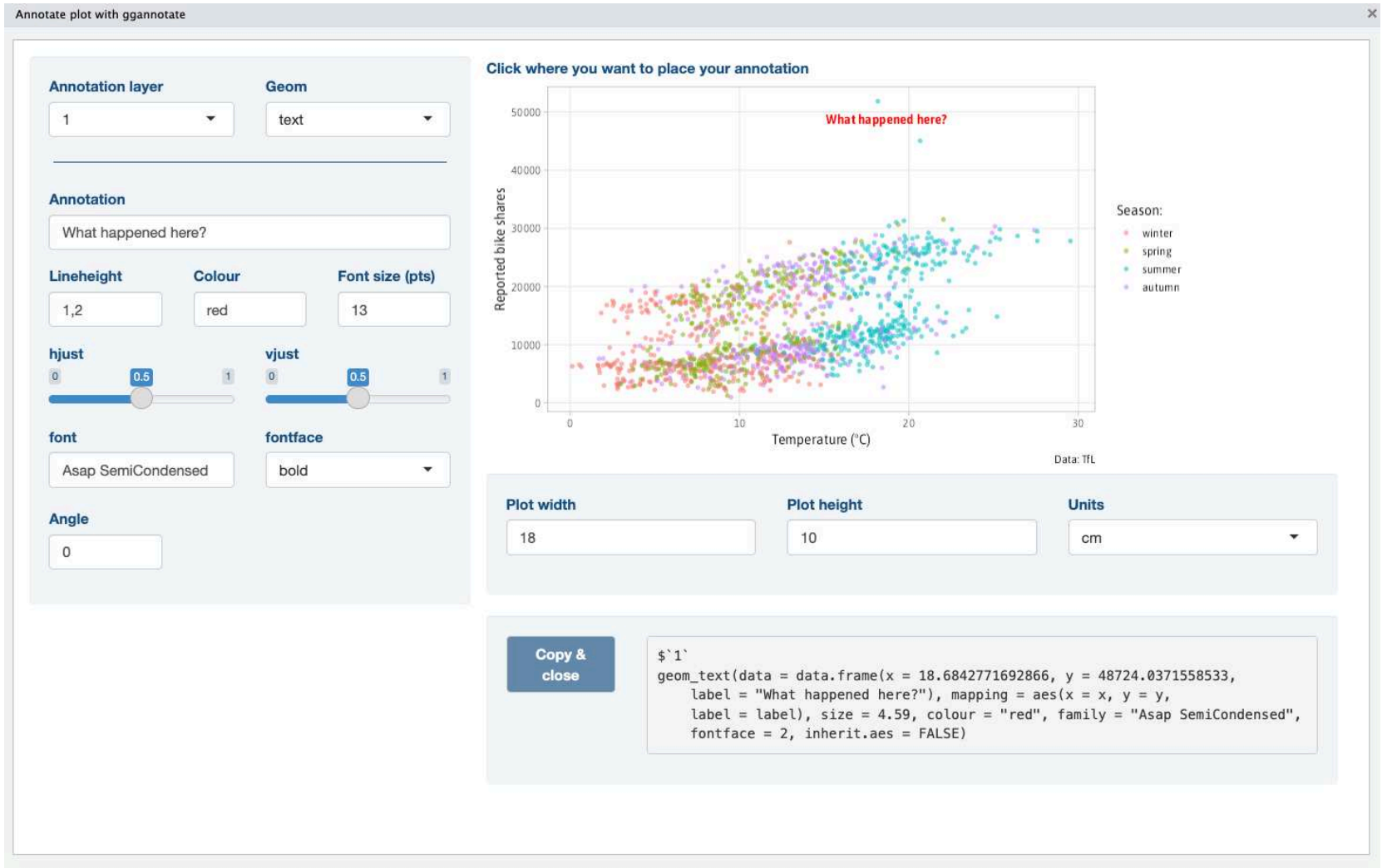
Add Multiple Text Annotations

```
ga +
  annotate(
    geom = "text",
    x = c(760, max(fannie_agg$avg_s
    y = c(250000, 10000),
    label = c("What happened here?'
    color = c("firebrick", "black")
    size = c(6, 3),
    family = c("Asap SemiCondensed'
    fontface = c("bold", "plain"),
    hjust = c(.5, 1)
  )
```



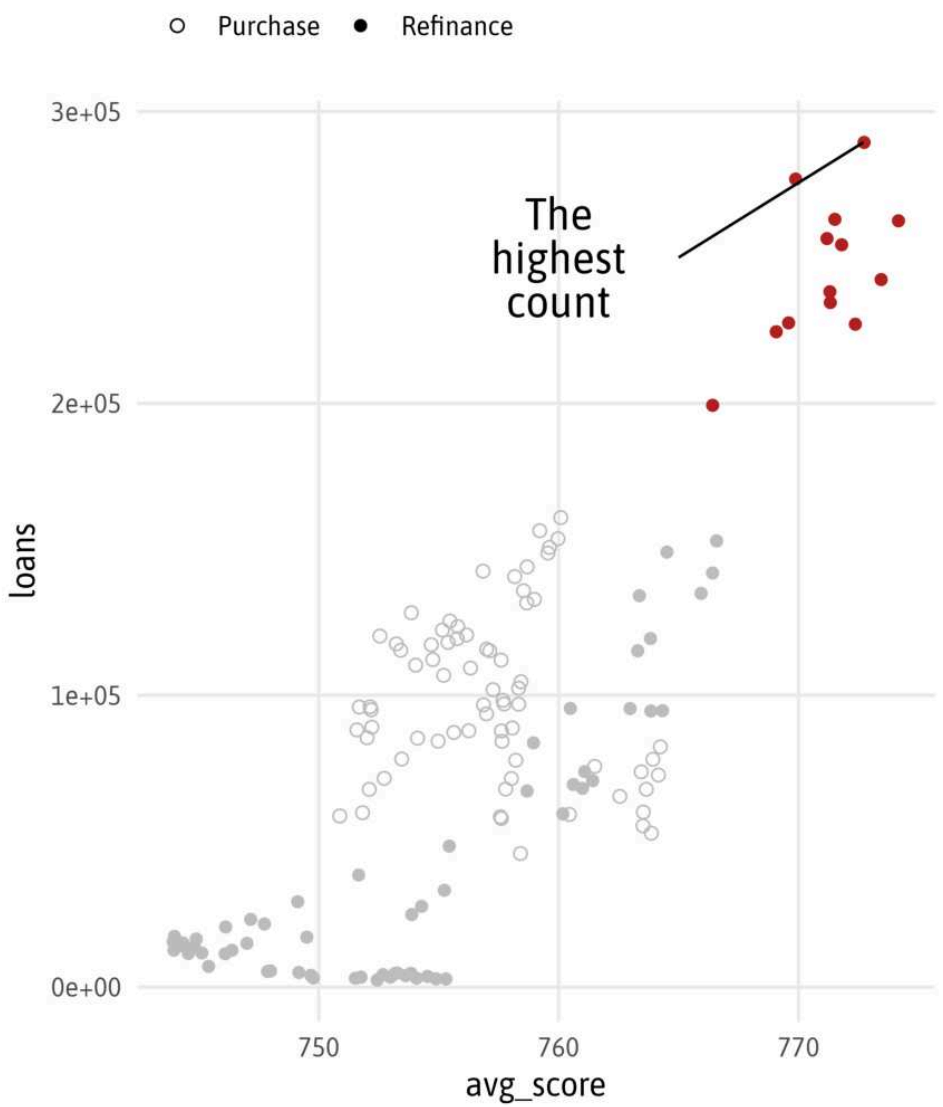
"Point'n'Click" Annotations

```
1  ggannotate::ggannotate(g)
```



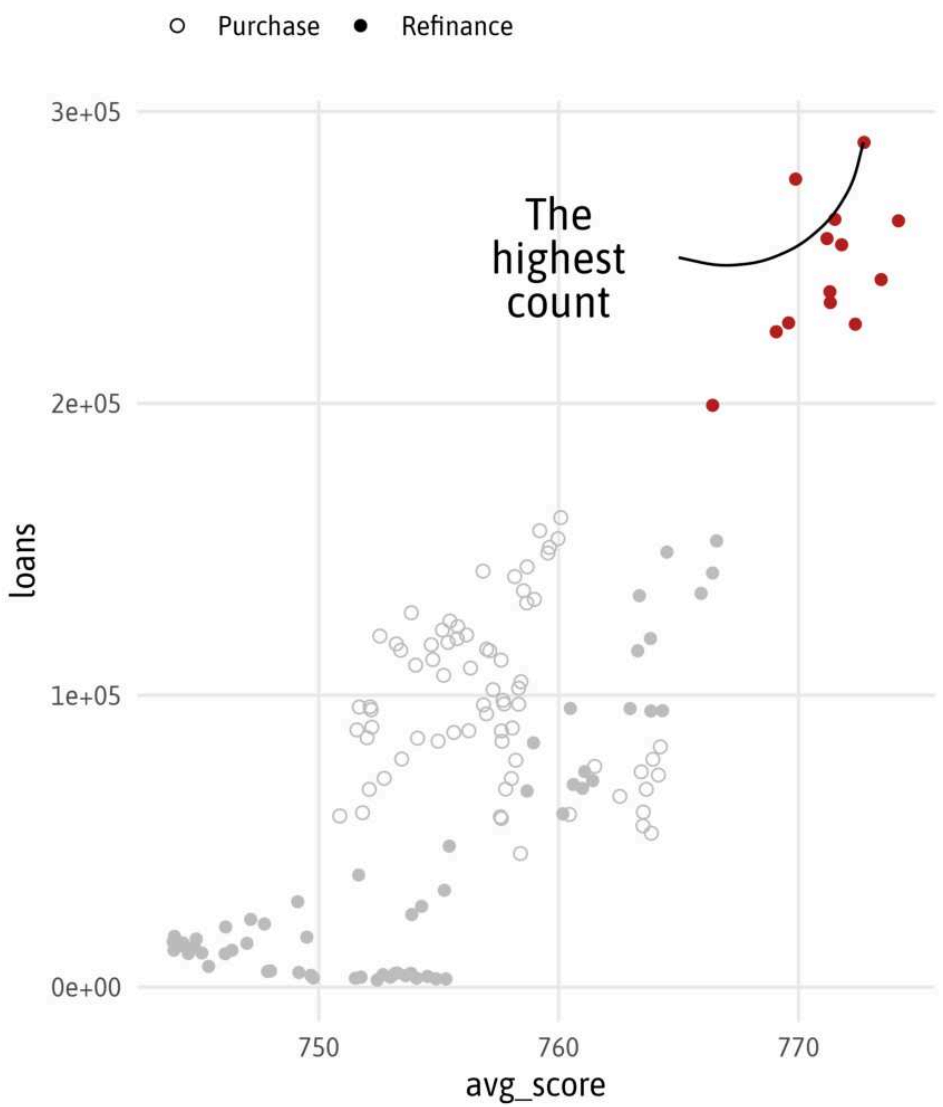
Add Lines

```
ga +
  annotate(
    geom = "text",
    x = 760,
    y = 250000,
    label = "The\nhighest\ncount",
    family = "Asap SemiCondensed",
    size = 6,
    lineheight =  .8
  ) +
  annotate(
    geom = "segment",
    x = 765,
    xend = 772.7,
    y = 250000,
    yend = 289435
  )
```



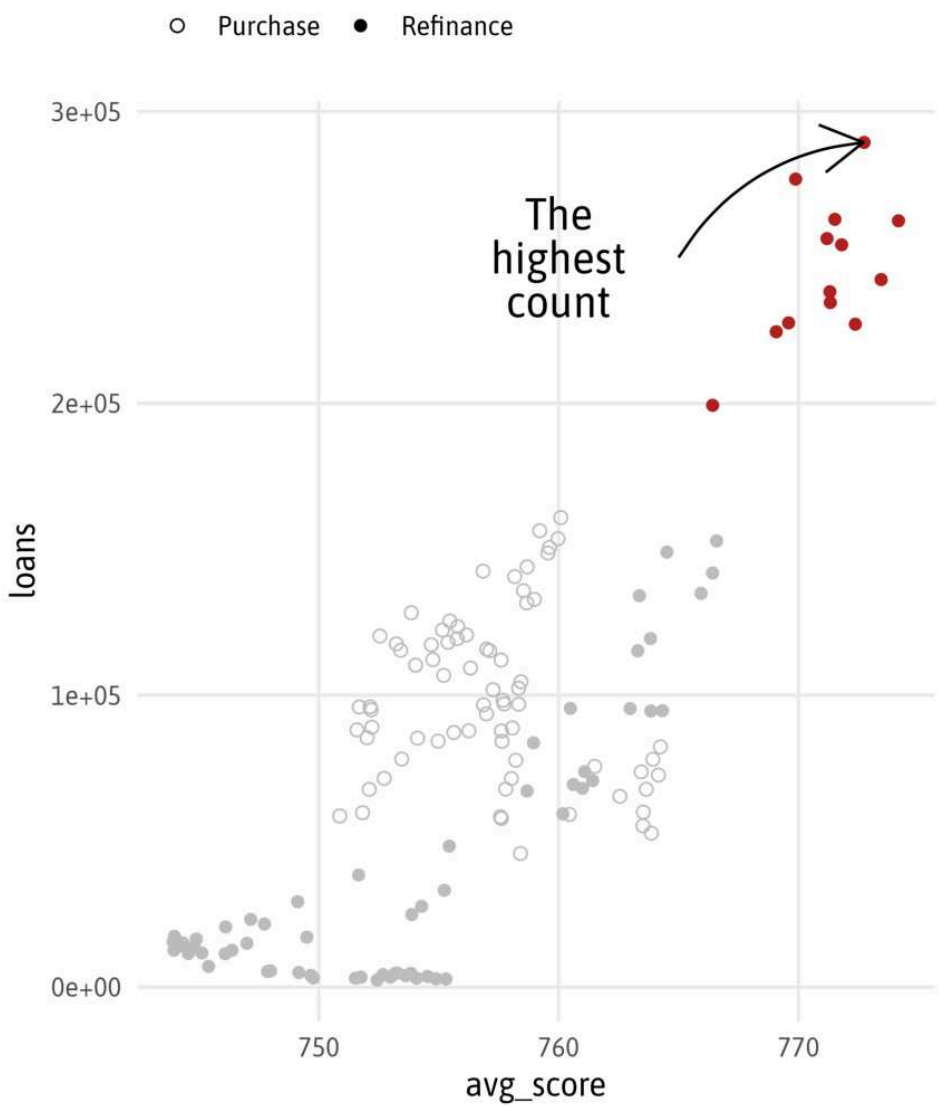
Add Lines

```
ga +
  annotate(
    geom = "text",
    x = 760,
    y = 250000,
    label = "The\nhighest\ncount",
    family = "Asap SemiCondensed",
    size = 6,
    lineheight =  .8
  ) +
  annotate(
    geom = "curve",
    x = 765,
    xend = 772.7,
    y = 250000,
    yend = 289435
  )
```



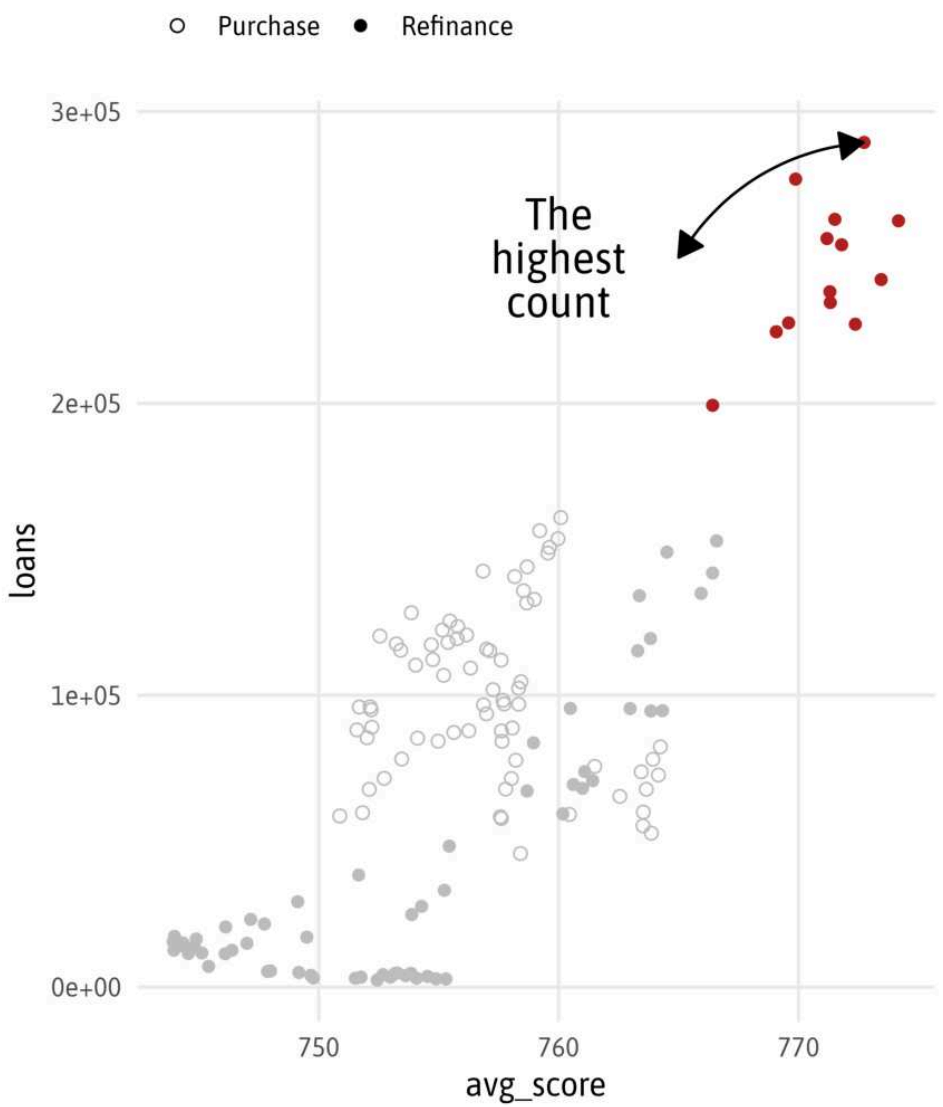
Add Arrows

```
ga +
  annotate(
    geom = "text",
    x = 760,
    y = 250000,
    label = "The\nhighest\ncount",
    family = "Asap SemiCondensed",
    size = 6,
    lineheight =  .8
  ) +
  annotate(
    geom = "curve",
    x = 765,
    xend = 772.7,
    y = 250000,
    yend = 289435,
    curvature = -.25,
    arrow = arrow()
  )
```



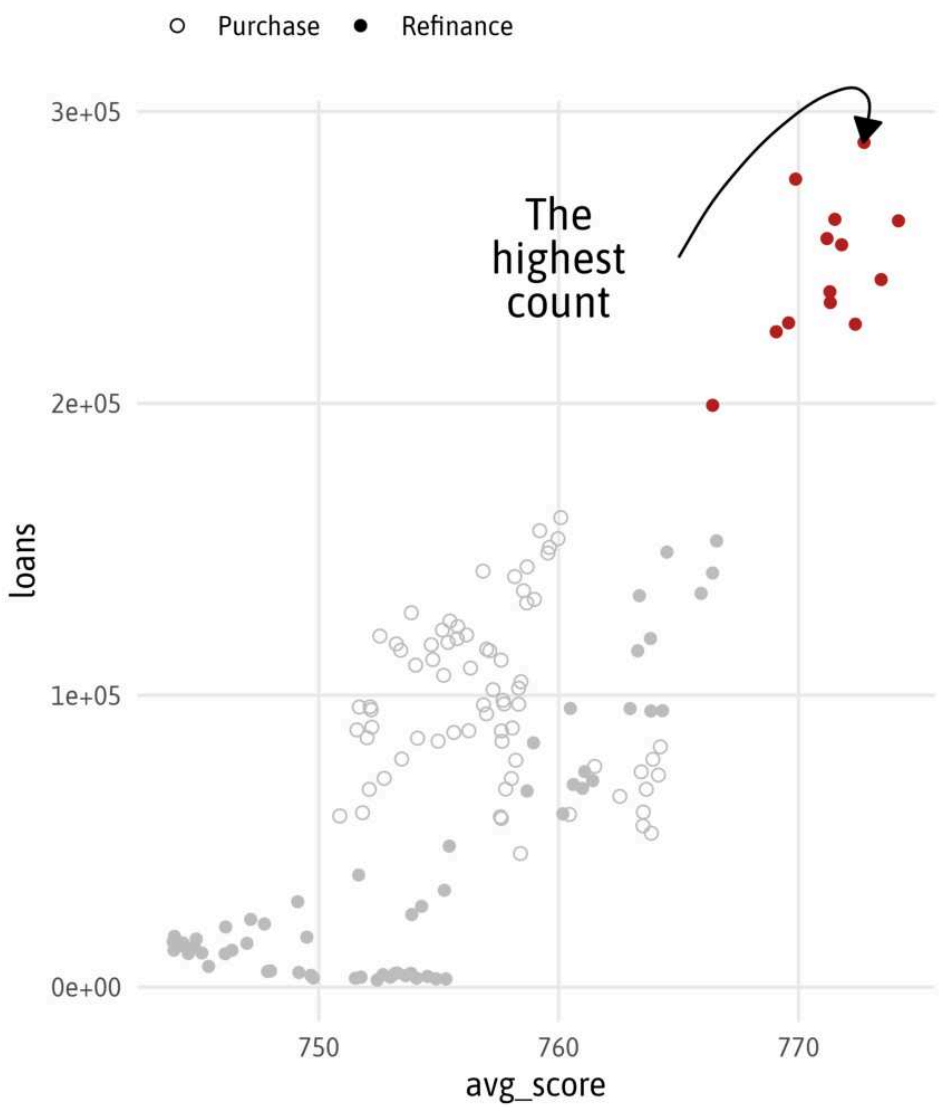
Add Arrows

```
ga +
  annotate(
    geom = "text",
    x = 760,
    y = 250000,
    label = "The\nhighest\ncount",
    family = "Asap SemiCondensed",
    size = 6,
    lineheight =  .8
  ) +
  annotate(
    geom = "curve",
    x = 765,
    xend = 772.7,
    y = 250000,
    yend = 289435,
    curvature = -.25,
    arrow = arrow(
      length = unit(10, "pt"),
      type = "closed",
      ends = "both"
    )
  )
```



Add Arrows

```
ga +
  annotate(
    geom = "text",
    x = 760,
    y = 250000,
    label = "The\nhighest\ncount",
    family = "Asap SemiCondensed",
    size = 6,
    lineheight =  .8
  ) +
  annotate(
    geom = "curve",
    x = 765,
    xend = 772.7,
    y = 250000,
    yend = 289435,
    curvature = -1,
    angle = -145,
    arrow = arrow(
      length = unit(10, "pt"),
      type = "closed"
    )
  )
```

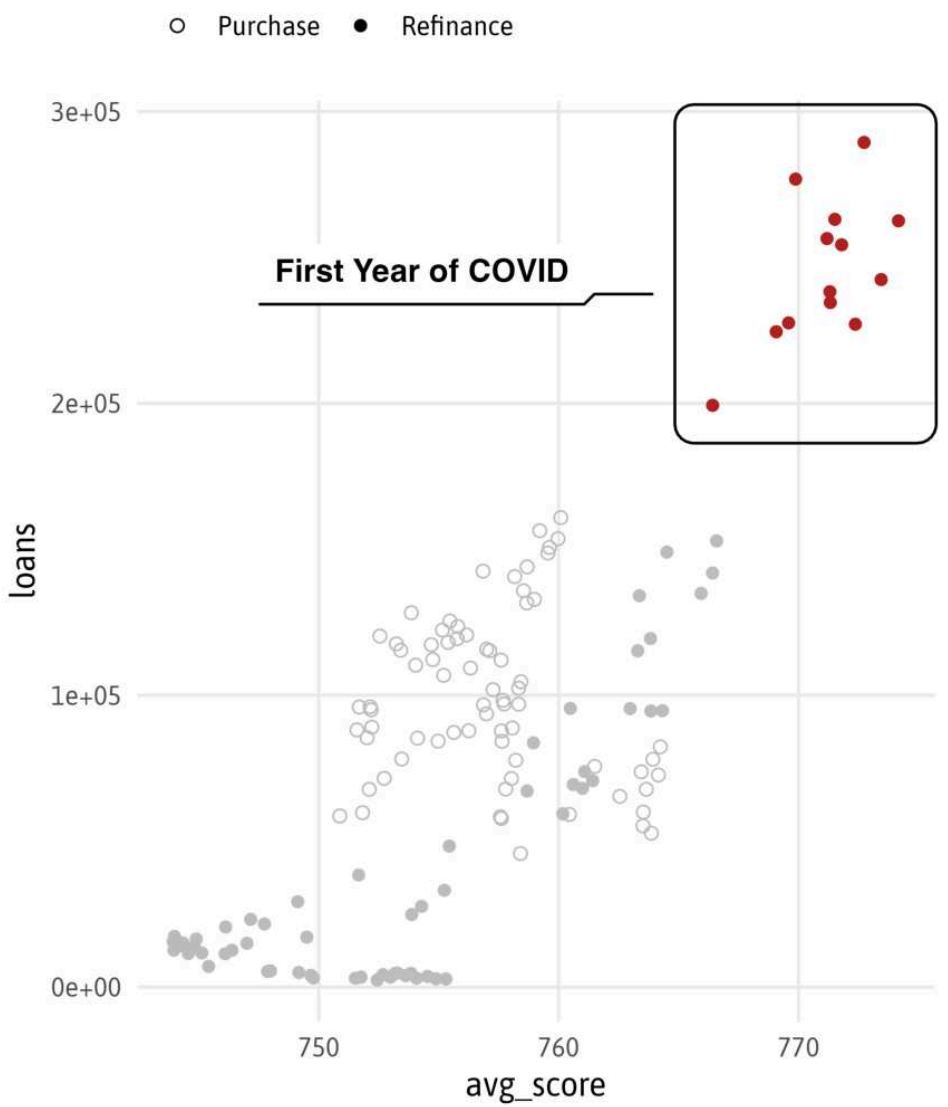


{ggforce}

Cédric Scherer Data Visualization & Information Design

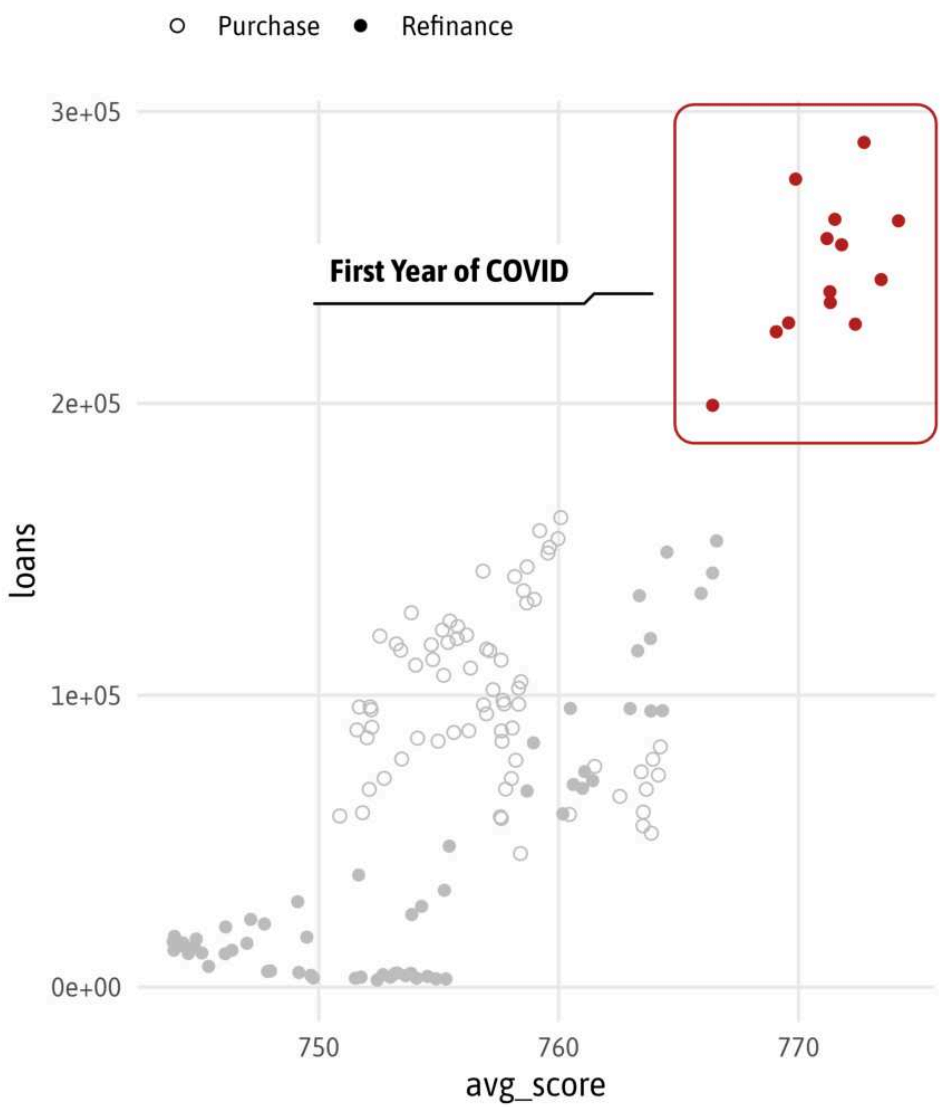
Annotations with {ggforce}

```
ga +
  ggforce::geom_mark_rect(
    aes(label = "First Year of COVI
        filter = loans > 190000)
  )
```



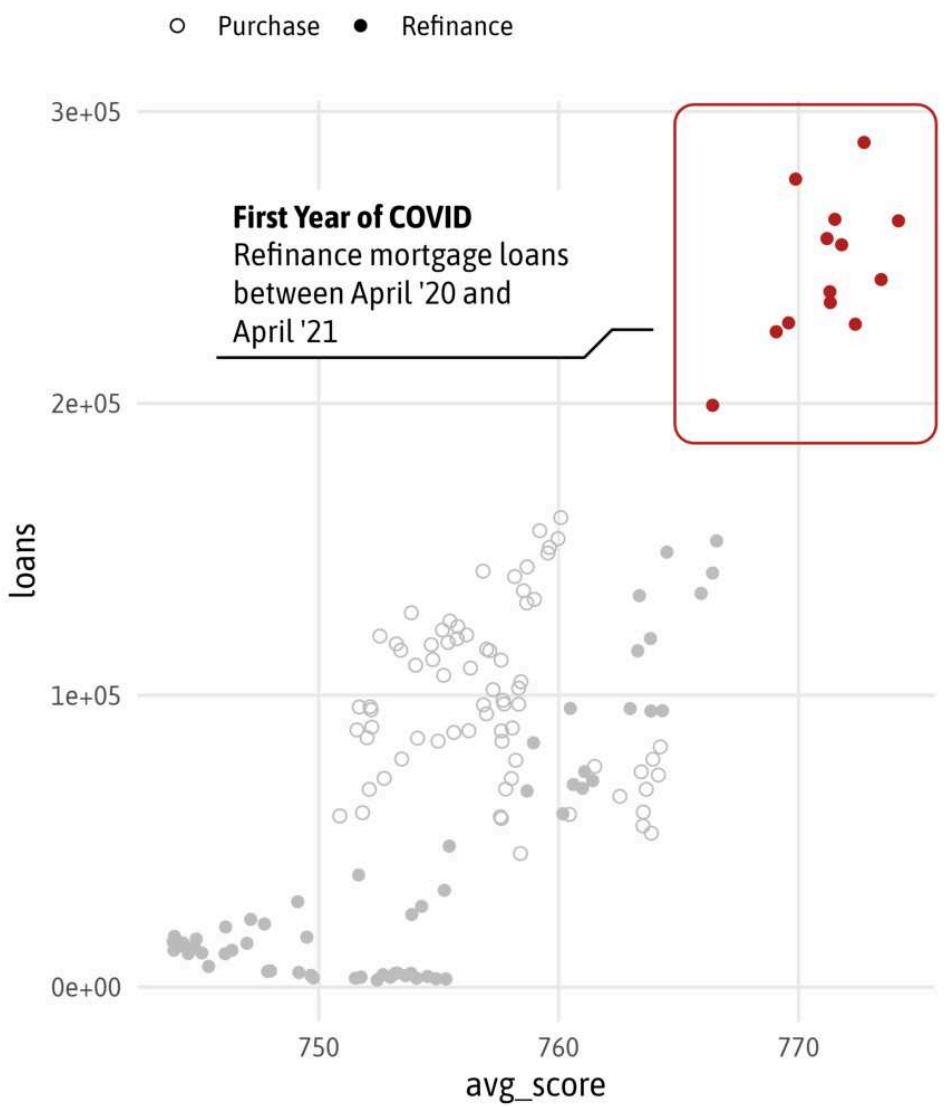
Annotations with {ggforce}

```
ga +
  ggforce::geom_mark_rect(
    aes(label = "First Year of COVI
        filter = loans > 190000),
    color = "firebrick",
    label.family = "Asap SemiConder
  )
```



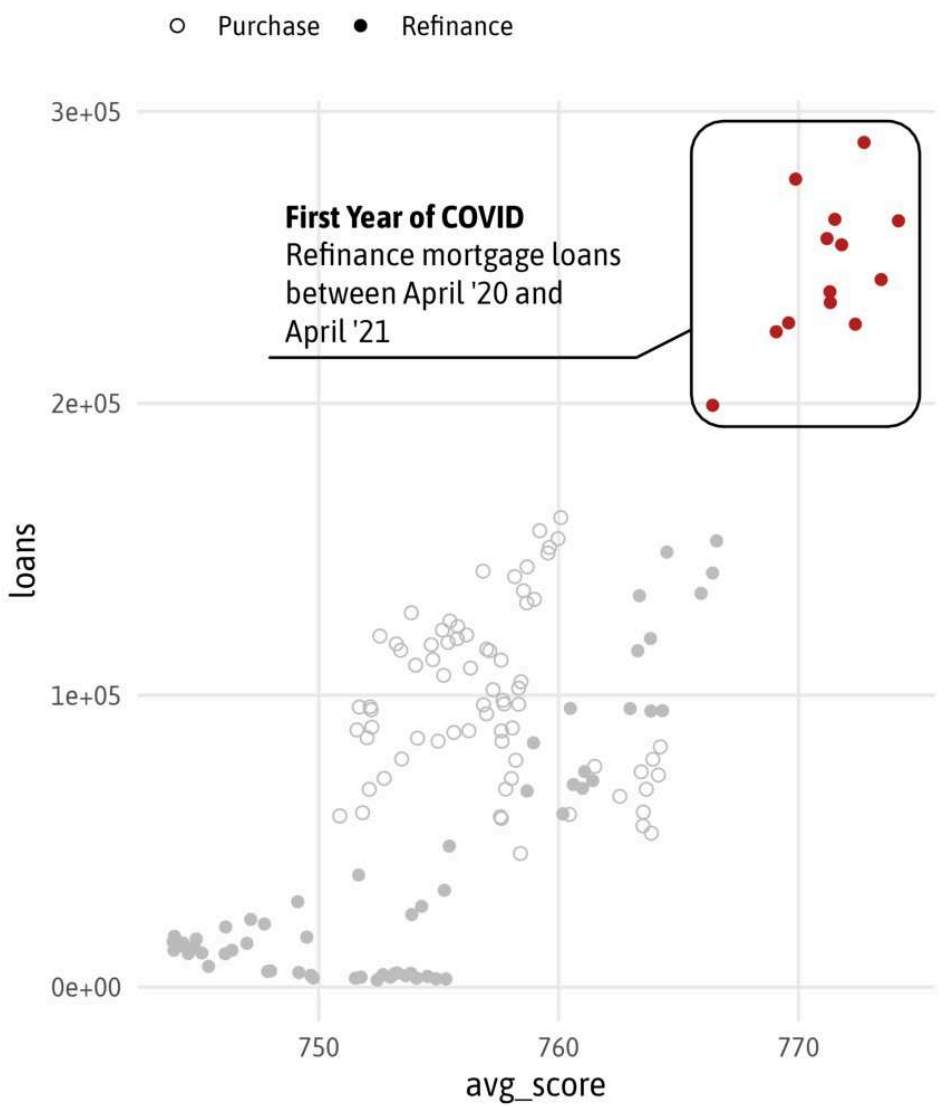
Annotations with {ggforce}

```
1 ga +
2   ggforce::geom_mark_rect(
3     aes(label = "First Year of COVI
4         filter = loans > 190000),
5     description = "Refinance mortga
6     color = "firebrick",
7     label.family = "Asap SemiConder
8   )
```



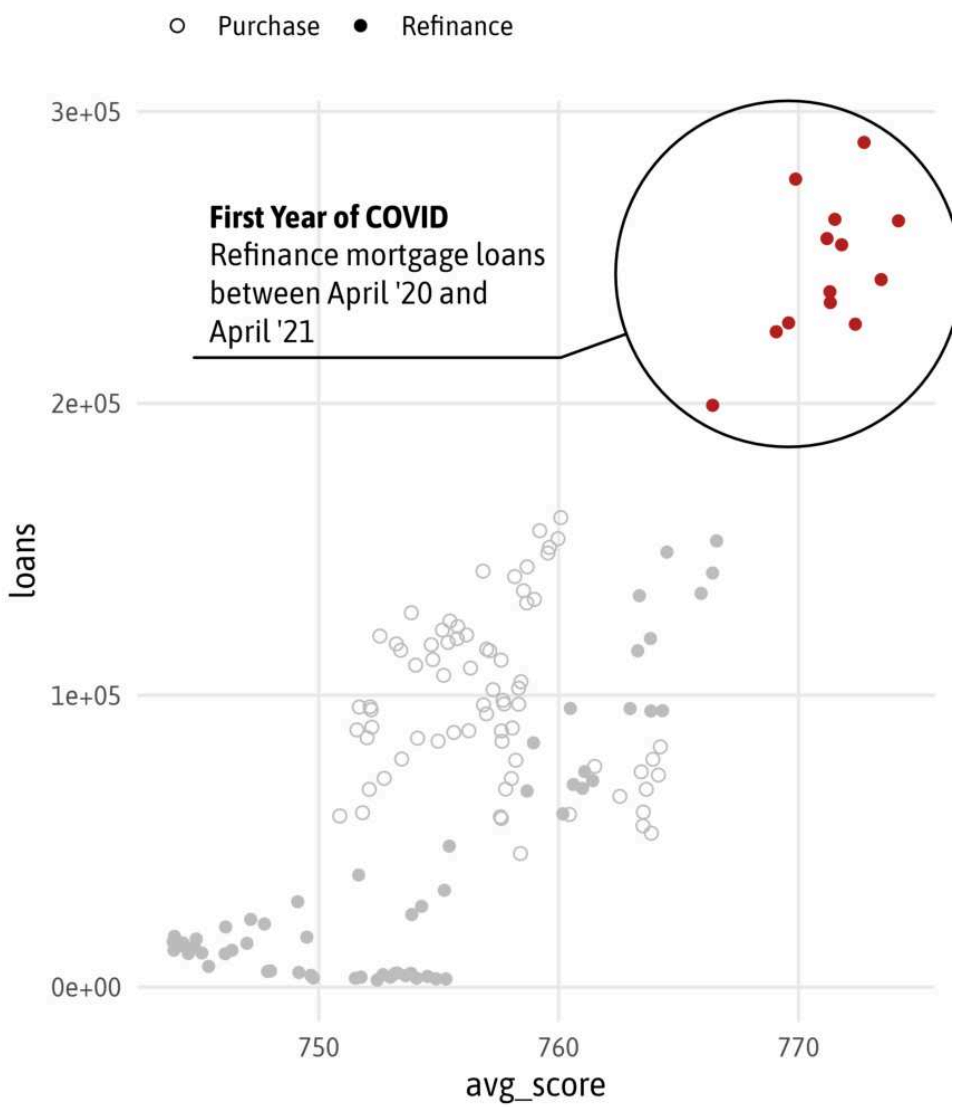
Annotations with {ggforce}

```
ga +
  ggforce::geom_mark_rect(
    aes(label = "First Year of COVI
        filter = loans > 190000),
    description = "Refinance mortga
    color = "black",
    label.family = "Asap SemiConder
    expand = unit(8, "pt"),
    radius = unit(12, "pt"),
    con.cap = unit(0, "pt"),
    label.buffer = unit(15, "pt"),
    con.type = "straight",
    label.fill = "transparent"
  )
```



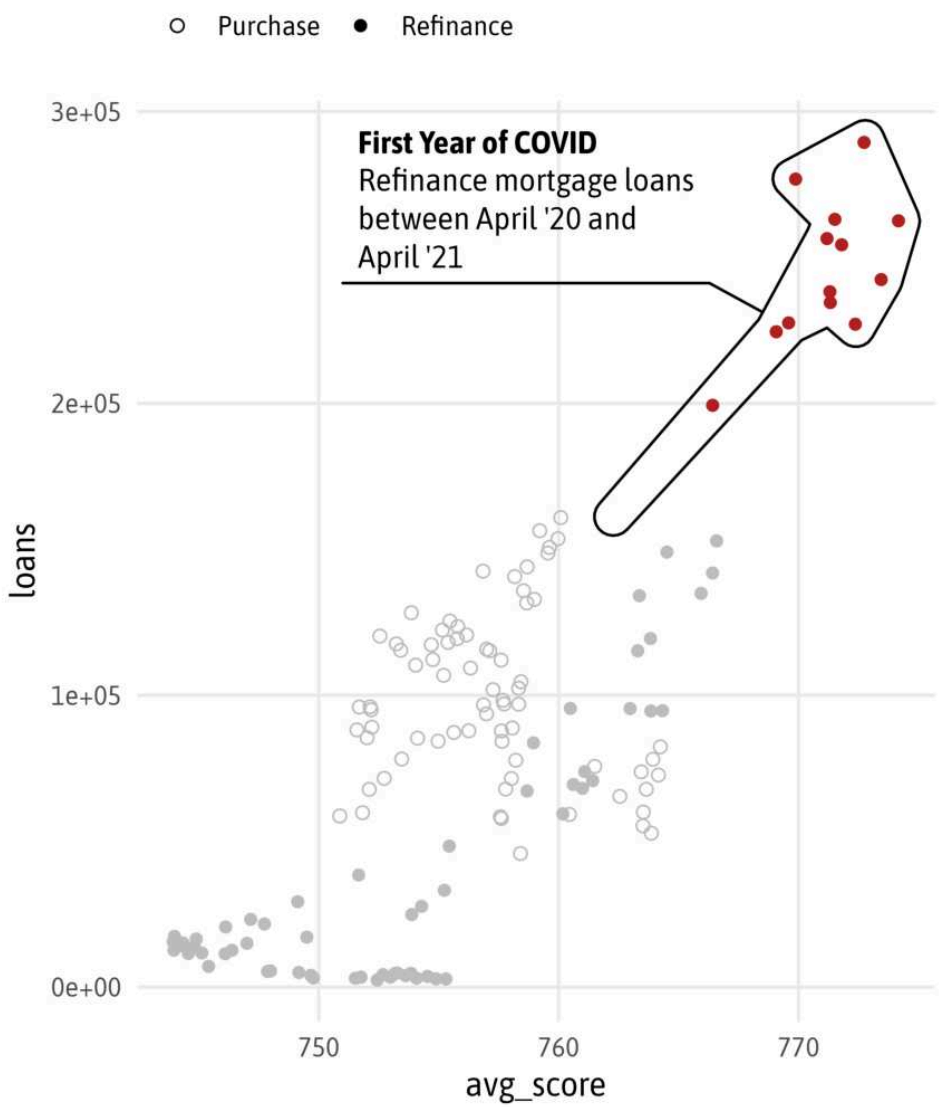
Annotations with {ggforce}

```
ga +
  ggforce::geom_mark_circle(
    aes(label = "First Year of COVI
        filter = loans > 190000),
    description = "Refinance mortga
    color = "black",
    label.family = "Asap SemiConder
    expand = unit(8, "pt"),
    con.cap = unit(0, "pt"),
    label.buffer = unit(15, "pt"),
    con.type = "straight",
    label.fill = "transparent"
  )
```



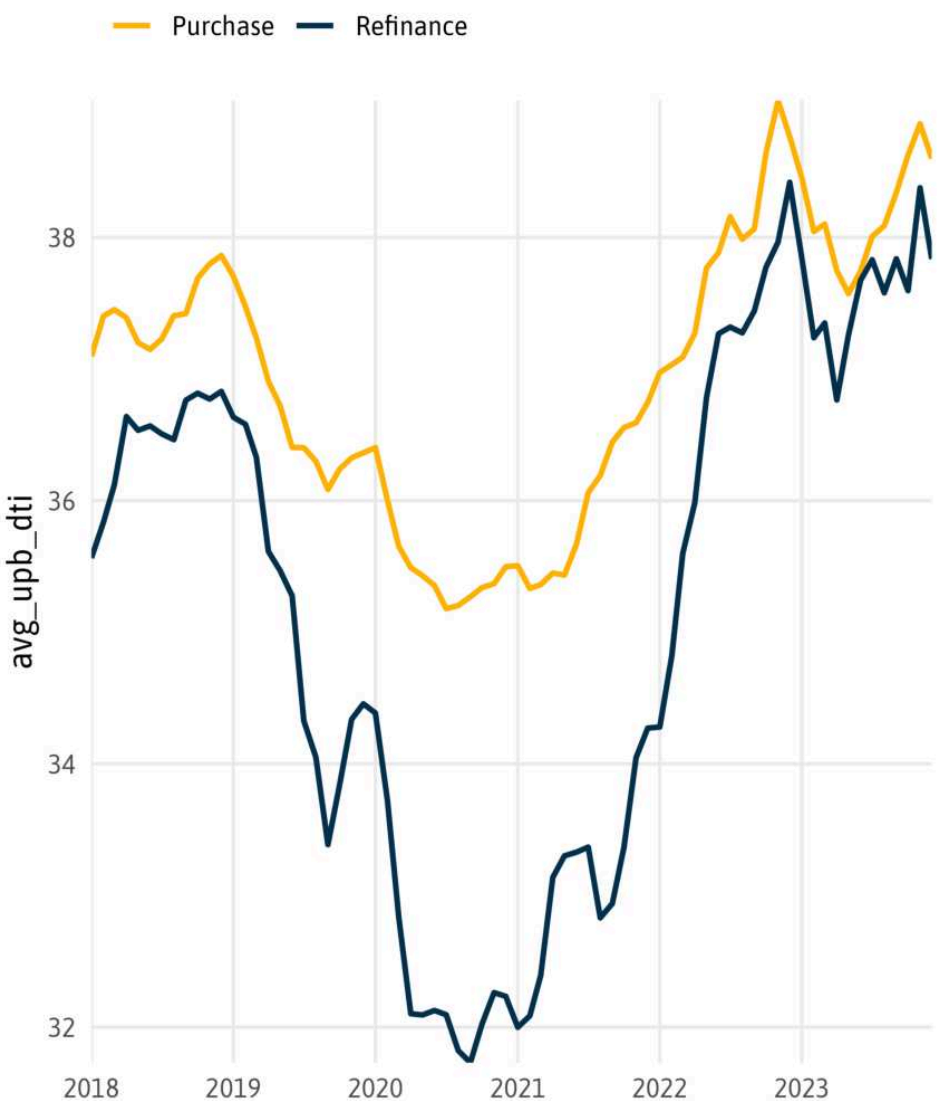
Annotations with {ggforce}

```
ga +
  ggforce::geom_mark_hull(
    aes(label = "First Year of COVI
        filter = loans > 190000),
    description = "Refinance mortga
    color = "black",
    label.family = "Asap SemiConder
    expand = unit(8, "pt"),
    con.cap = unit(0, "pt"),
    label.buffer = unit(15, "pt"),
    con.type = "straight",
    label.fill = "transparent"
  )
```



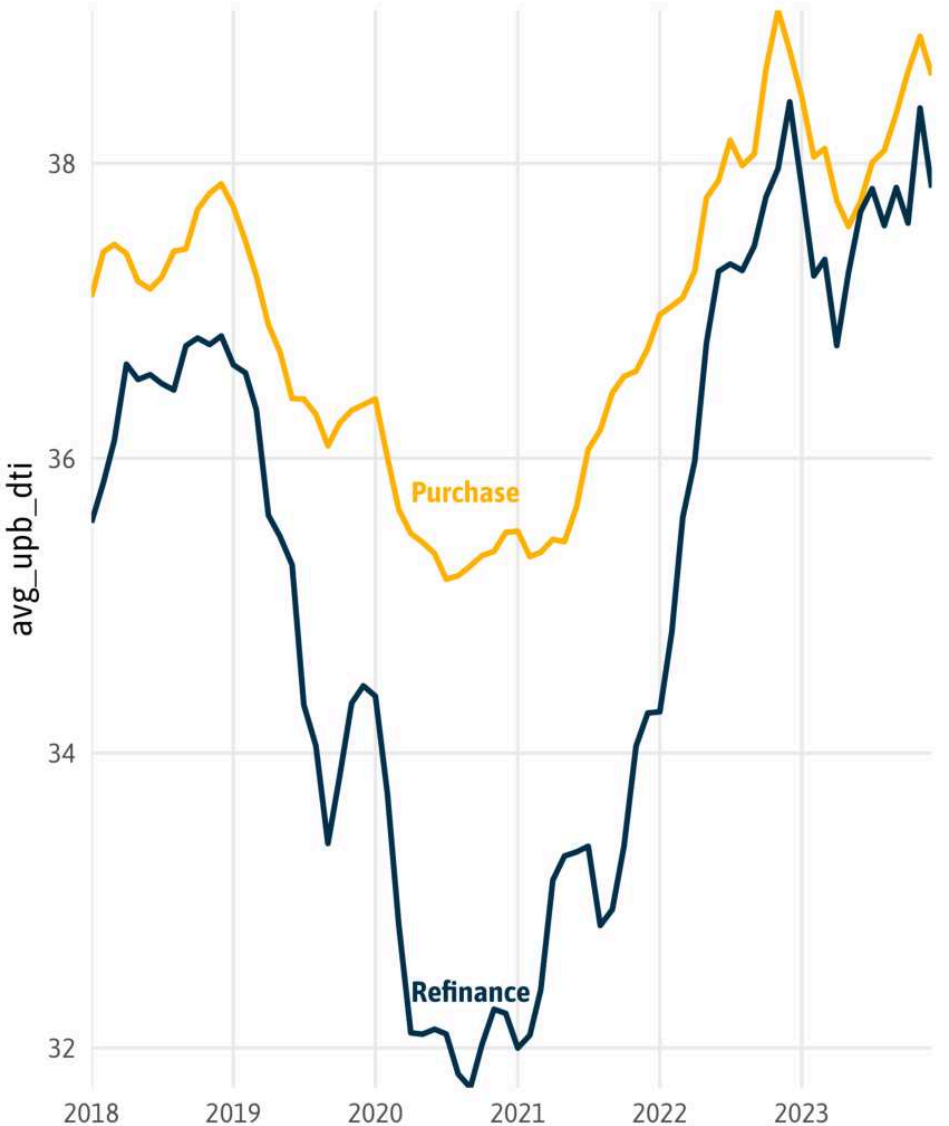
Annotating Line Charts

```
ggplot(
    fannie_agg,
    aes(x = orig_date, y = avg_upb_
        color = purpose,
        group = purpose)
  ) +
  geom_line(linewidth = 1) +
  coord_cartesian(expand = FALSE) +
  scale_color_manual(
    values = c("#FFB306", "#05314D'
    name = NULL
  ) +
  labs(x = NULL)
```



Annotating Line Charts

```
ggplot(
    fannie_agg,
    aes(x = orig_date, y = avg_upb_
        color = purpose,
        group = purpose)
  ) +
  geom_line(linewidth = 1) +
  geom_text(
    data = filter(fannie_agg, orig_
    aes(label = purpose),
    family = "Asap SemiCondensed",
    fontface = "bold",
    vjust = -1.5,
    hjust = 0
  ) +
  coord_cartesian(expand = FALSE) +
  scale_color_manual(
    values = c("#FFB306", "#05314D"
    guide = "none"
  ) +
  labs(x = NULL)
```

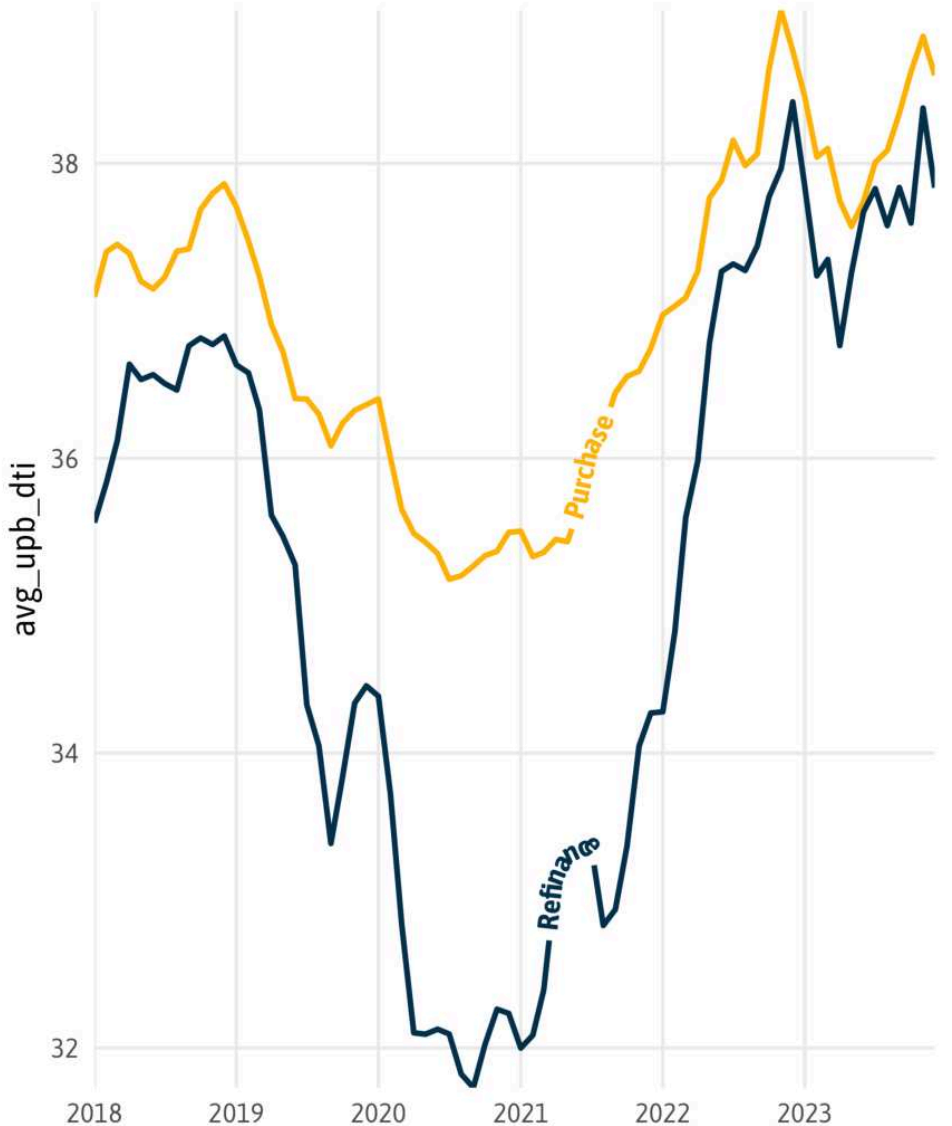


{geomtextpath}

Cédric Scherer Data Visualization & Information Design

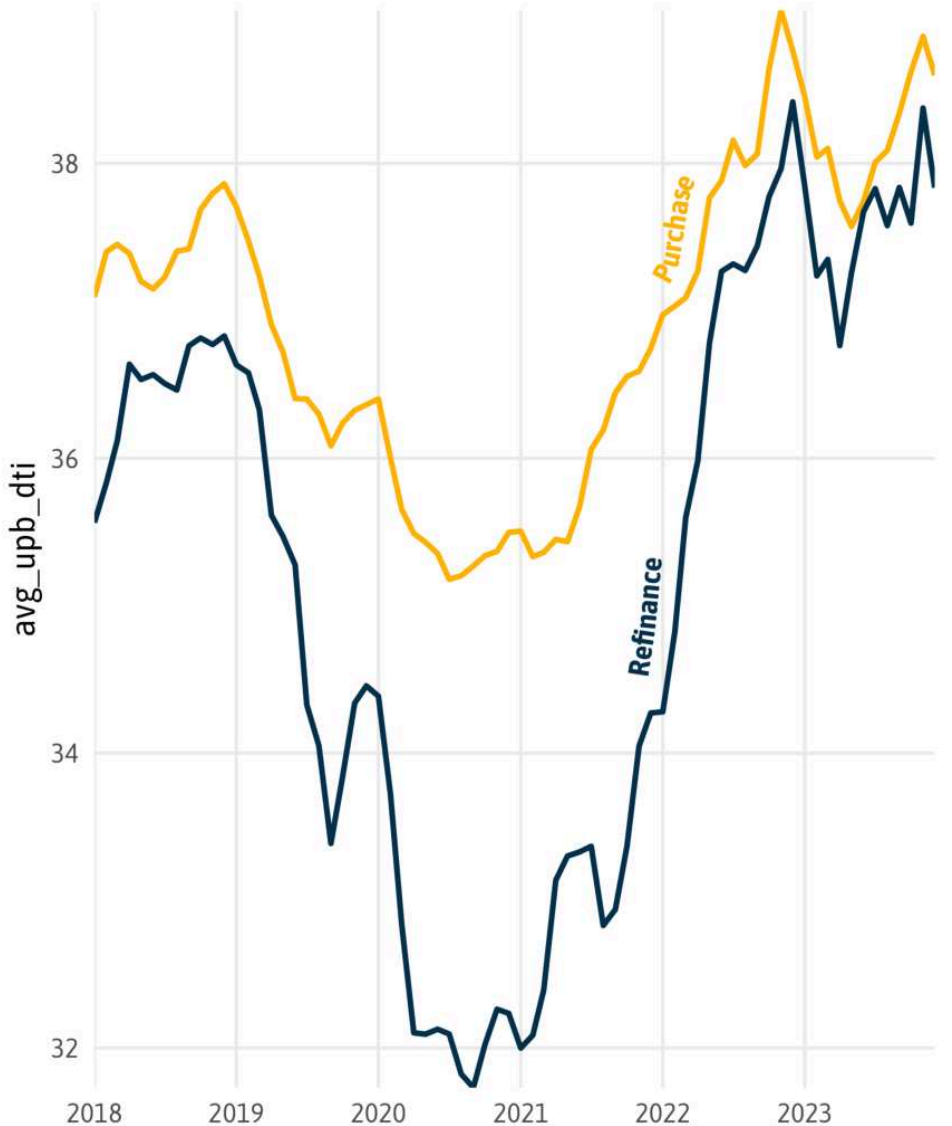
Annotations with {geomtextpath}

```
ggplot(
    fannie_agg,
    aes(x = orig_date, y = avg_upb_
        color = purpose,
        group = purpose)
  ) +
  geomtextpath::geom_textline(
    aes(label = purpose),
    linewidth = 1,
    family = "Asap SemiCondensed",
    fontface = "bold"
  ) +
  coord_cartesian(expand = FALSE) +
  scale_color_manual(
    values = c("#FFB306", "#05314D"
    guide = "none"
  ) +
  labs(x = NULL)
```



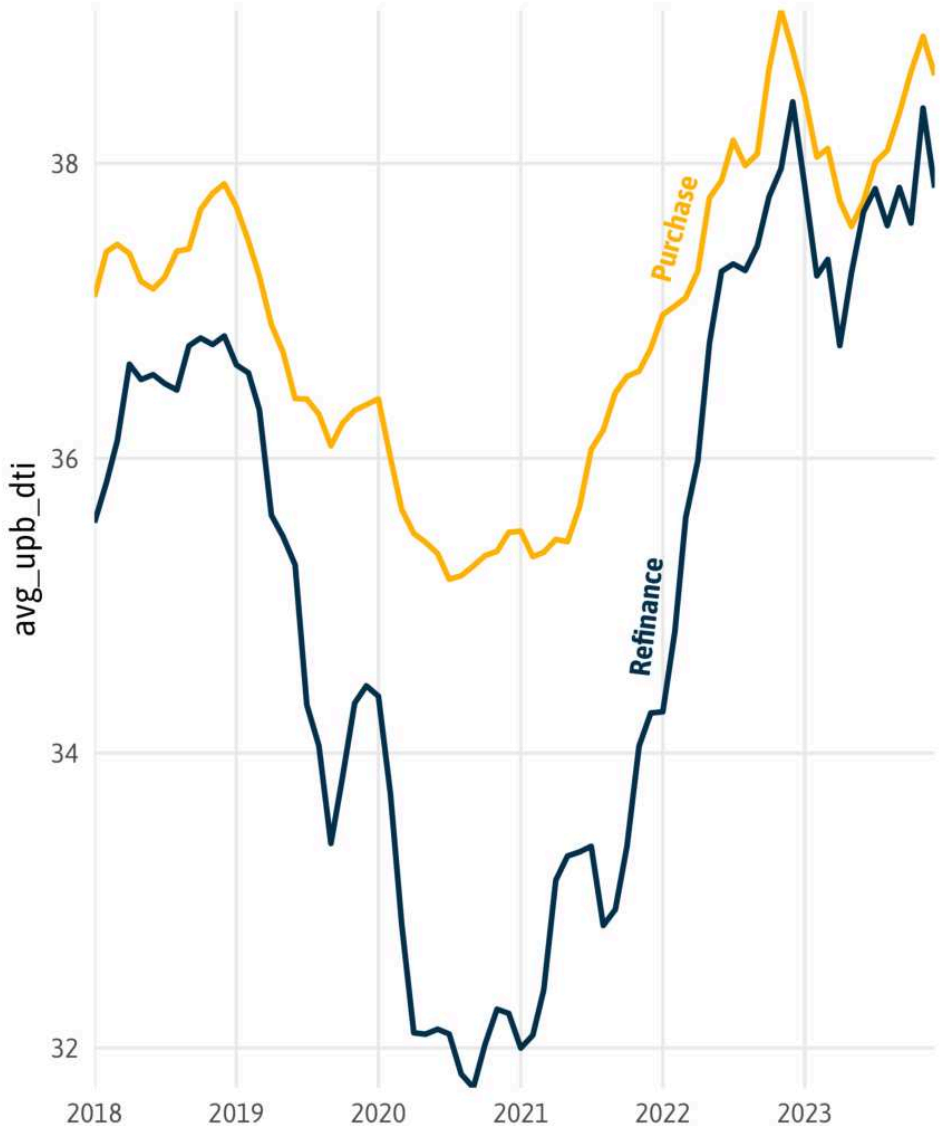
Annotations with {geomtextpath}

```
ggplot(
    fannie_agg,
    aes(x = orig_date, y = avg_upb_
        color = purpose,
        group = purpose)
  ) +
  geomtextpath::geom_textline(
    aes(label = purpose),
    linewidth = 1,
    family = "Asap SemiCondensed",
    fontface = "bold",
    vjust = -.3,
    hjust = .63
  ) +
  coord_cartesian(expand = FALSE) +
  scale_color_manual(
    values = c("#FFB306", "#05314D"
    guide = "none"
  ) +
  labs(x = NULL)
```



Annotations with {geomtextpath}

```
ggplot(
    fannie_agg,
    aes(x = orig_date, y = avg_upb_
        color = purpose,
        group = purpose)
  ) +
  geomtextpath::geom_textline(
    aes(label = purpose),
    linewidth = 1,
    family = "Asap SemiCondensed",
    fontface = "bold",
    vjust = -.3,
    hjust = .63,
    straight = TRUE
  ) +
  coord_cartesian(expand = FALSE) +
  scale_color_manual(
    values = c("#FFB306", "#05314D"
    guide = "none"
  ) +
  labs(x = NULL)
```



Text Rendering

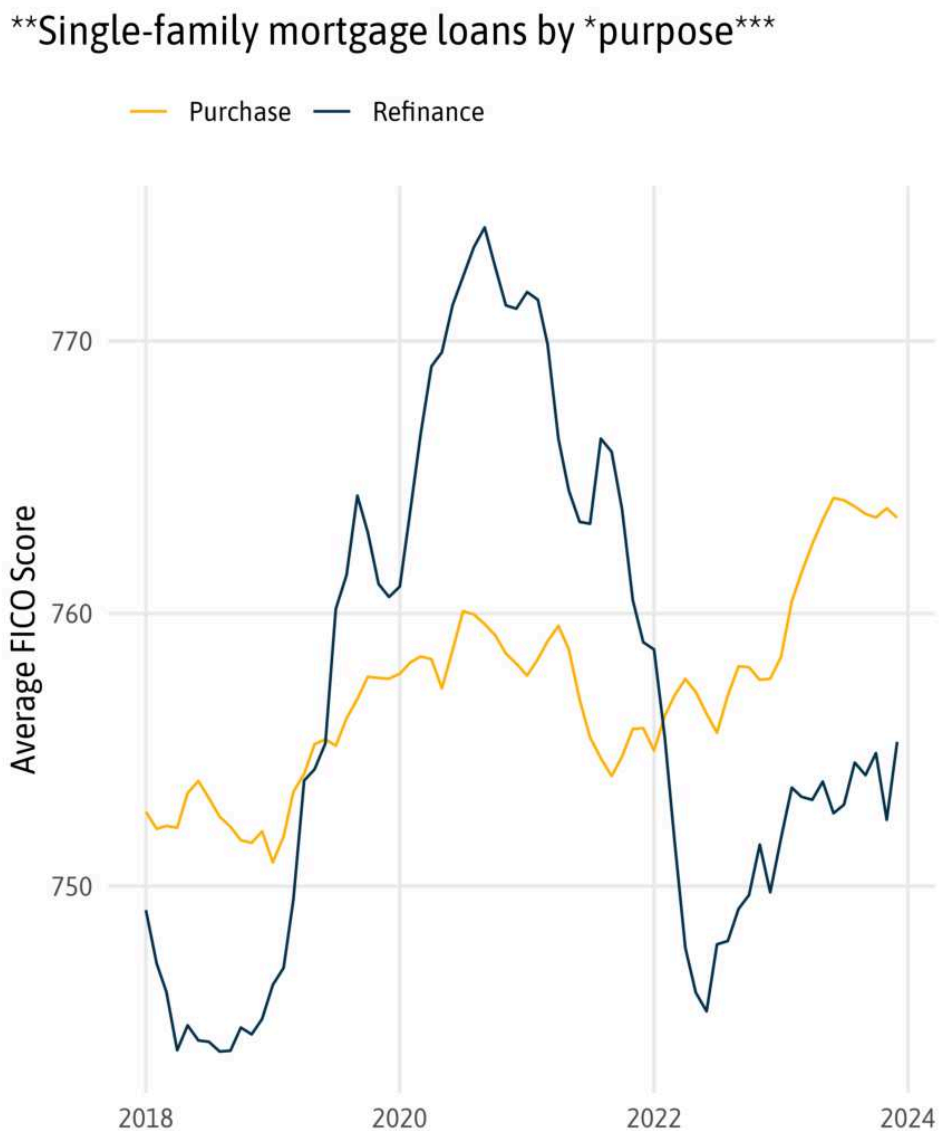
Cédric Scherer Data Visualization & Information Design

{ggtext}

Cédric Scherer Data Visualization & Information Design

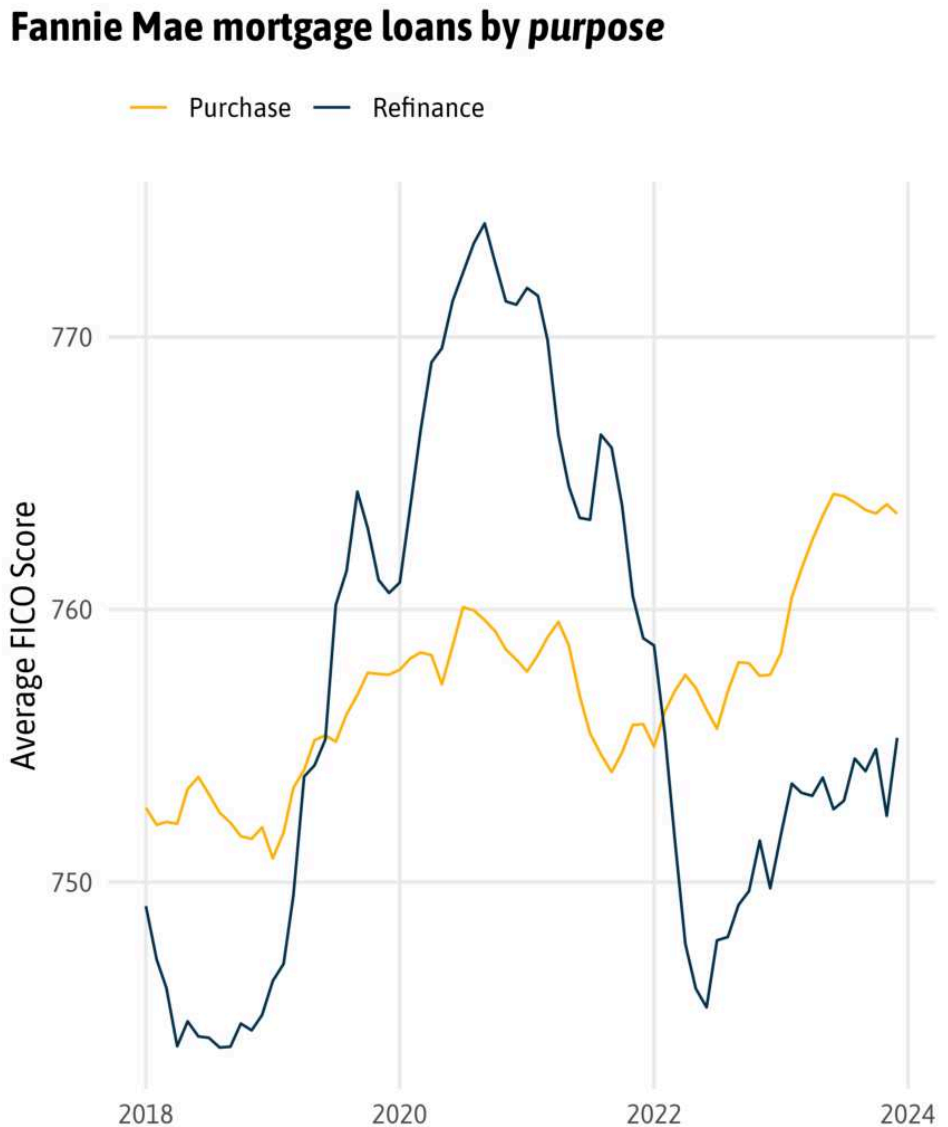
Styling Labels with {ggtext}

```
g +
  ggtitle("**Single-family mortgage
```



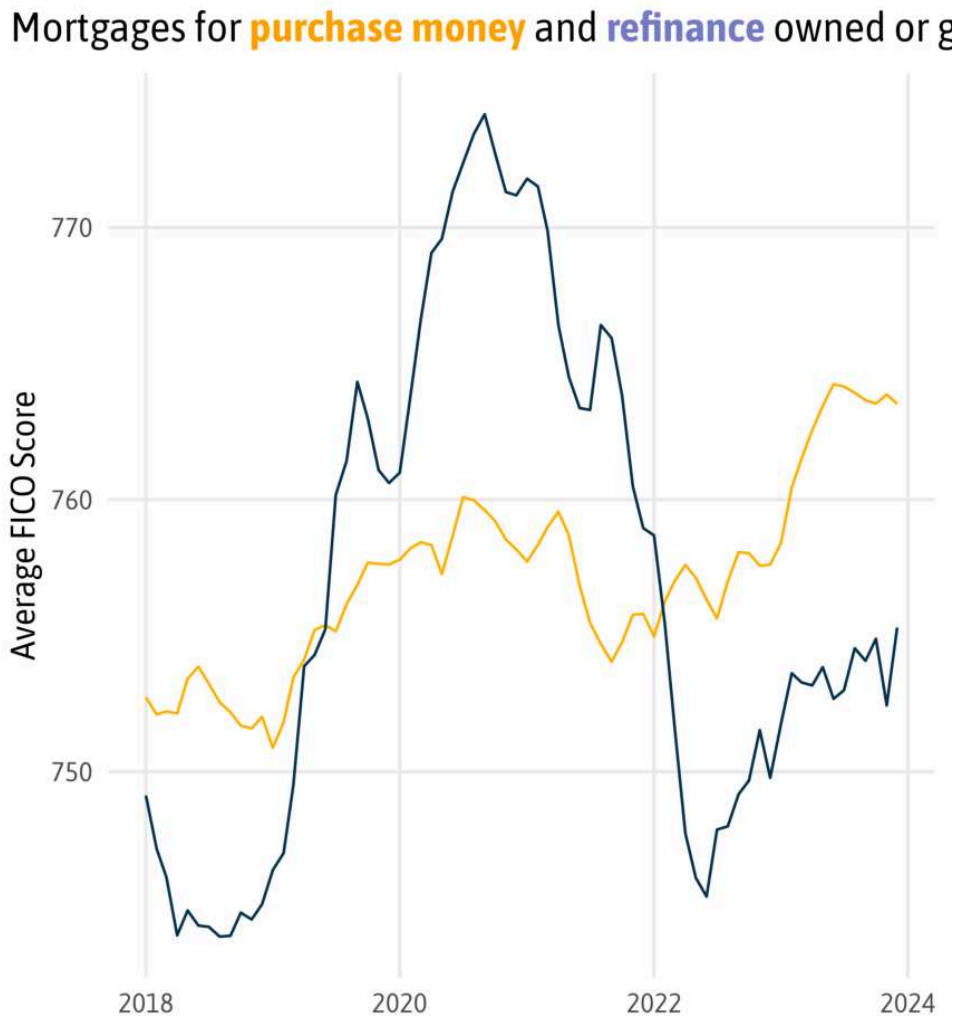
Styling Labels with {ggtext}

```
g +
  ggtitle("**Fannie Mae mortgage lo
  theme(
    plot.title = ggtext::element_ma
  )
```



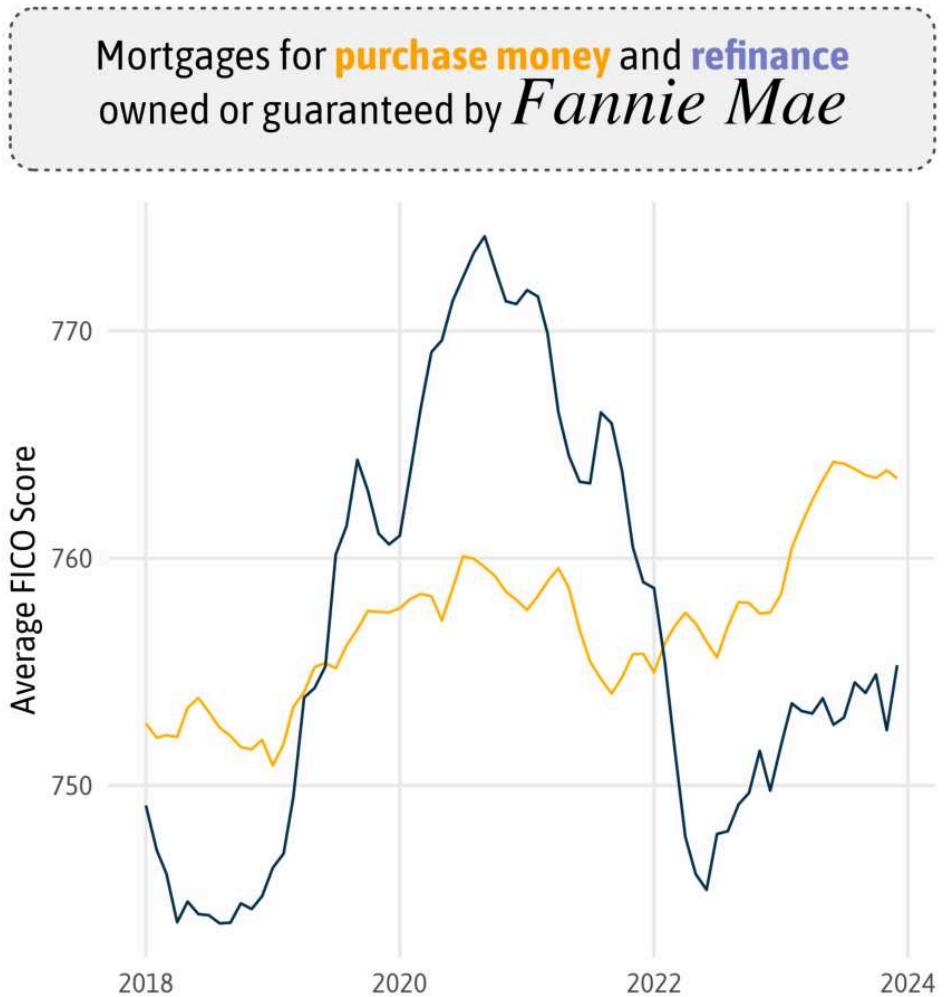
Styling Labels with {ggtext}

```
g +
  ggtitle("Mortgages for <b style='
  theme(
    plot.title = ggtext::element_ma
    legend.position = "none"
  )
```



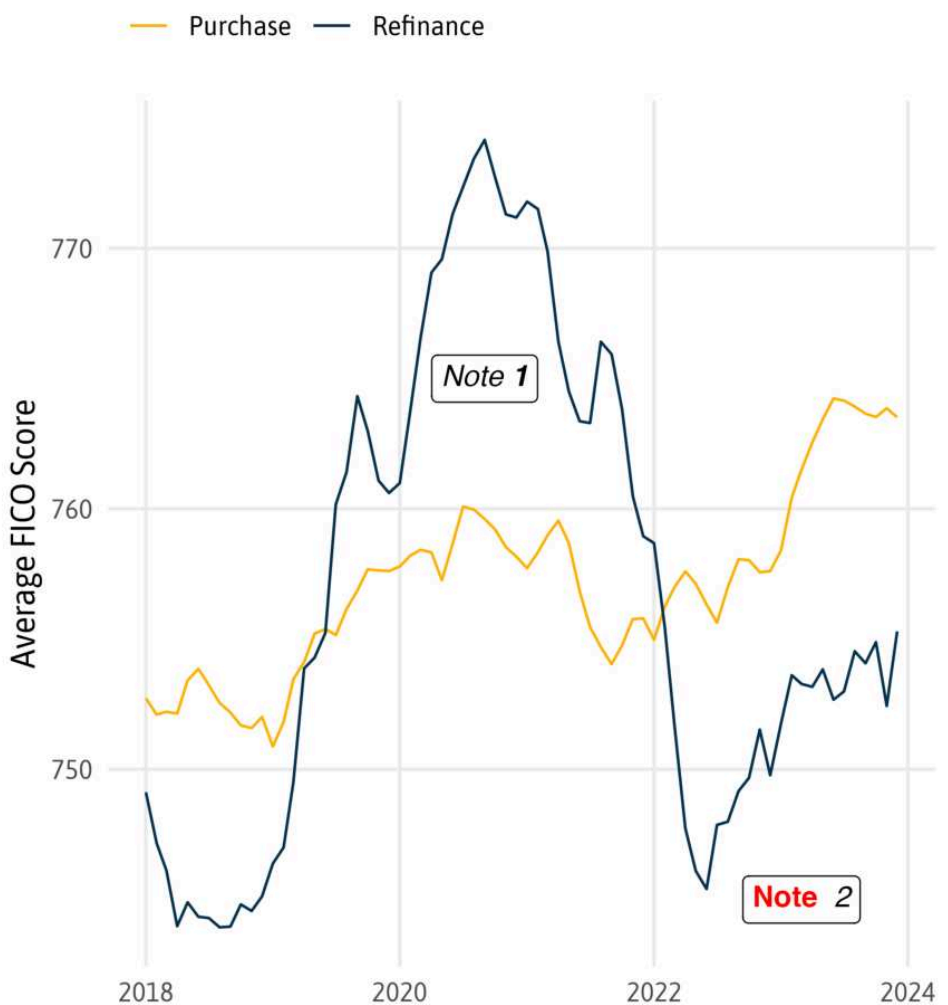
Styling Labels with {ggtext}

```
g +
  ggtitle("Mortgages for <b style='
  theme(
    plot.title =
      ggtext::element_textbox_simpl
        margin = margin(t = 12, b =
        padding = margin(rep(12, 4)
        fill = "grey95",
        box.colour = "grey30",
        linetype = "13",
        r = unit(9, "pt"),
        halign = .5,
        lineheight = 1
      ),
    legend.position = "none"
  )
```



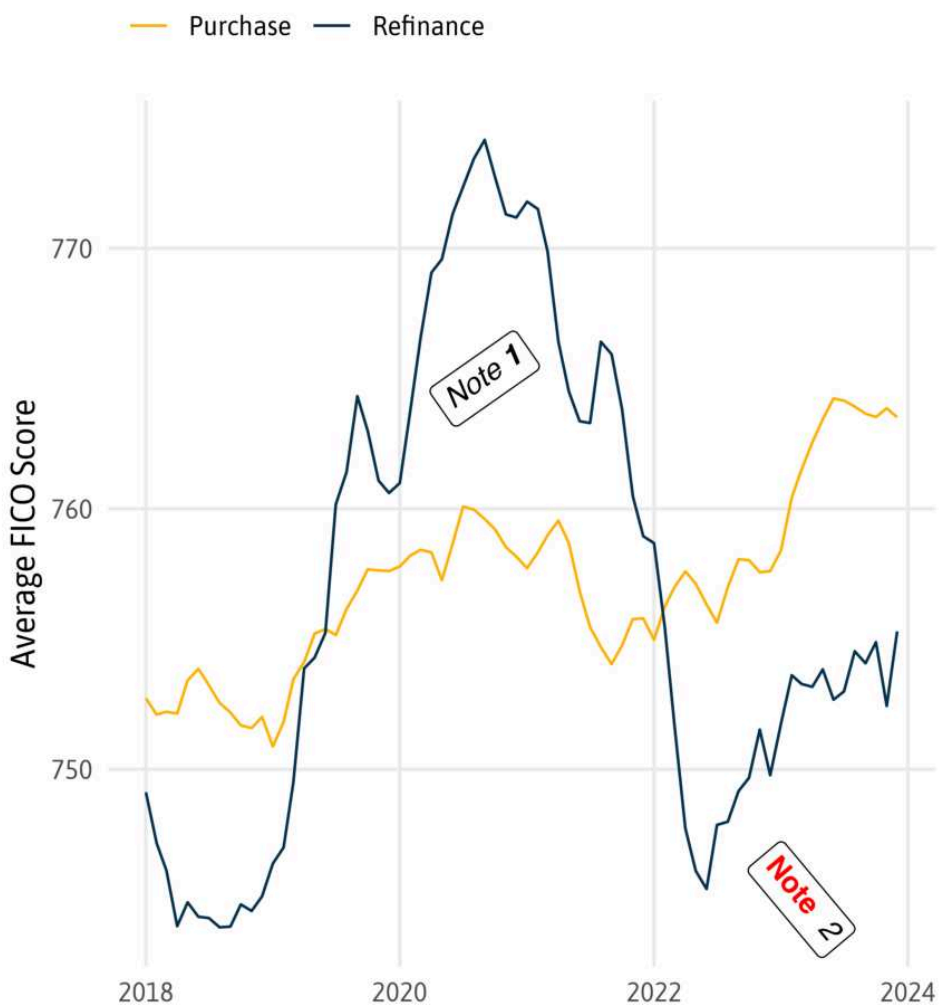
Adding Annotations with {ggtext}

```
g +
  ggtext::geom_richtext(
    data = data.frame(
      orig_date = as_date(c("2020-0
      avg_score = c(765, 745),
      lab = c("*Note **1***",
              "<b style='color:red;
    ),
    aes(label = lab),
    color = "black"
  )
```



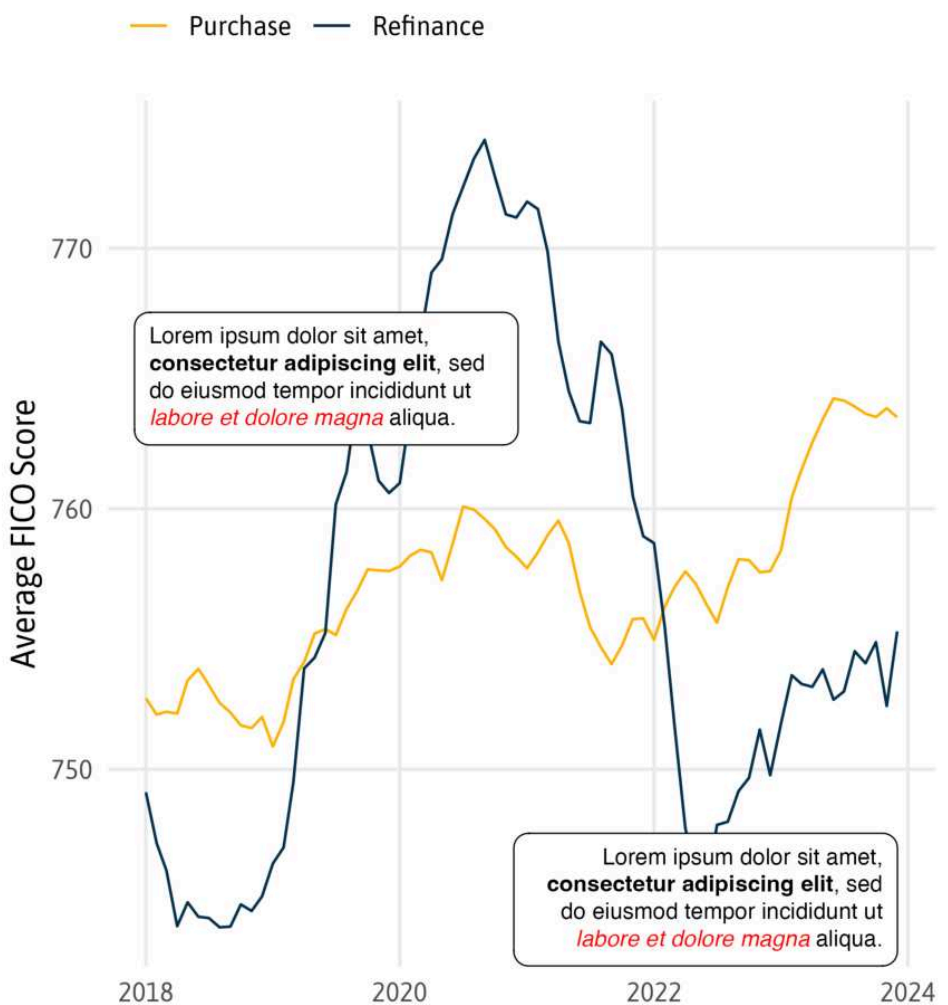
Adding Annotations with {ggtext}

```
g +
  ggtext::geom_richtext(
    data = data.frame(
      orig_date = as_date(c("2020-0
      avg_score = c(765, 745),
      lab = c("*Note **1***",
              "<b style='color:red;
      angle = c(35, 310)
    ),
    aes(label = lab,
        angle = angle),
    color = "black"
  )
```



Adding Annotations with {ggtext}

```
g +
  ggtext::geom_textbox(
    data = data.frame(
      orig_date = as_date(c("2018-0
      avg_score = c(765, 745),
      lab = "Lorem ipsum dolor sit
    ),
    aes(label = lab),
    color = "black",
    size = 3,
    halign = c(0, 1),
    hjust = c(.25, .75)
  )
```

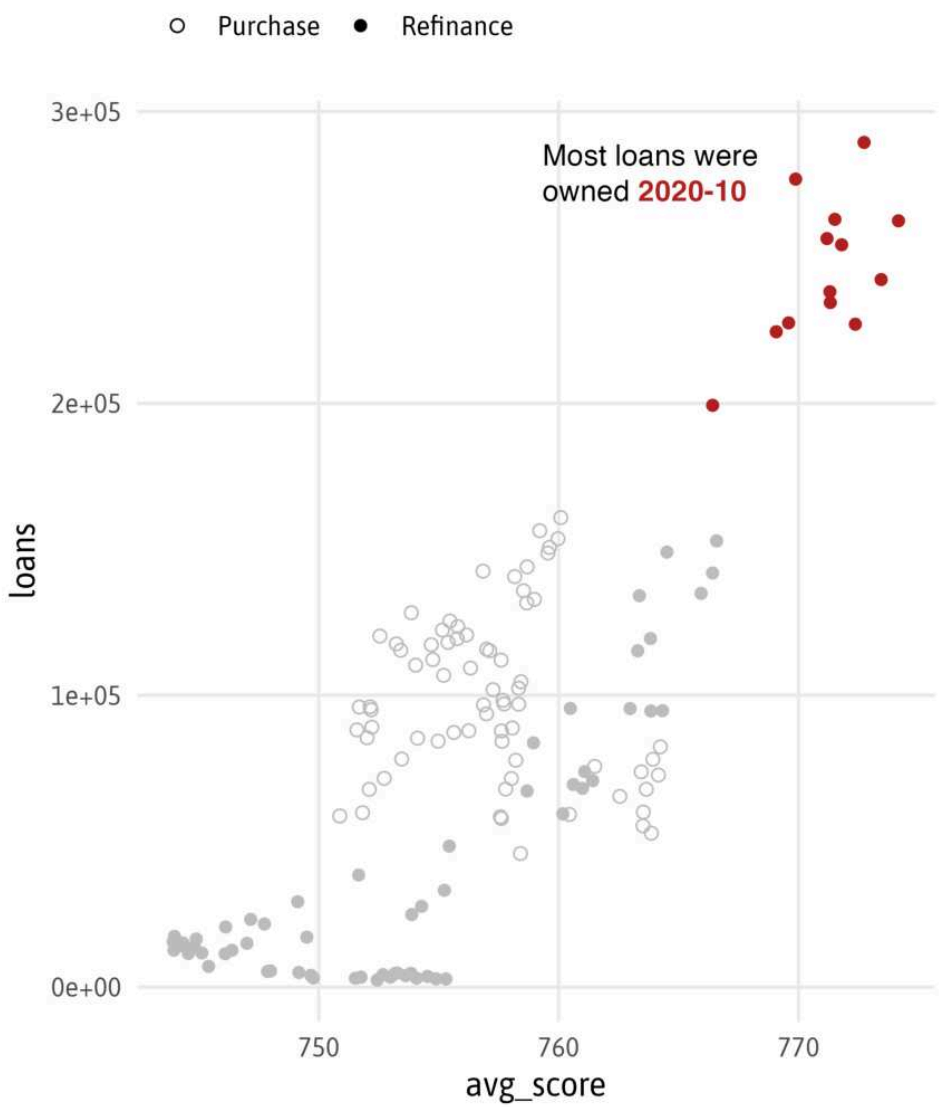


{marquee}

Cédric Scherer Data Visualization & Information Design

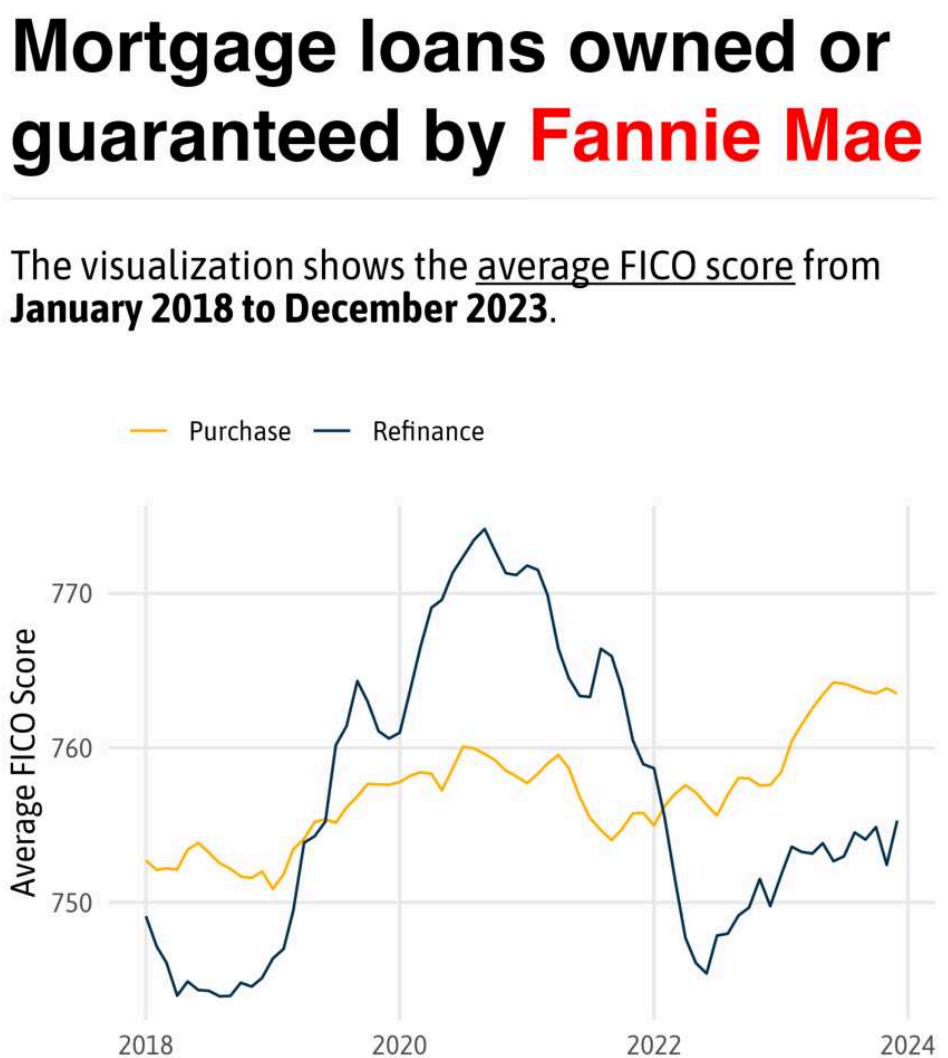
Formatted Text with {marquee}

```
ga +
  marquee::geom_marquee(
    data = filter(fannie_agg, loans
    aes(label = paste0(
      "Most loans were
      owned {.firebrick **", format
    )),
    color = "black",
    hjust = .75,
    vjust = 1
  )
```



Formatted Labels with {marquee}

```
md <-
"## Mortgage loans owned or guarant
The visualization shows the _averag

g +
  ggtitle(md) +
  theme(
    plot.title = marquee::element_m
      width = unit(1, "npc")
    )
  )
```



Formatted Labels with {marquee}

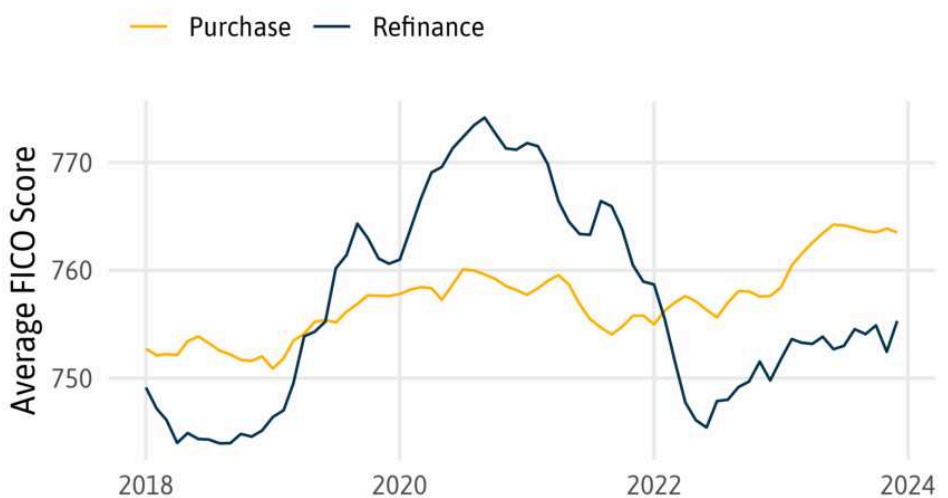
```
md <-
"One can even add code chunks:

    text <- \"markdown **text**\"
    marquee_grob(text)

... and even lists:

1. this
1. is
1. awesome!
"

g +
  ggtitle(md) +
  theme(
    plot.title = marquee::element_m
      width = unit(1, "npc")
    )
  )
```

One can even add code chunks:

```
text <- "markdown **text**"
marquee_grob(text)
```

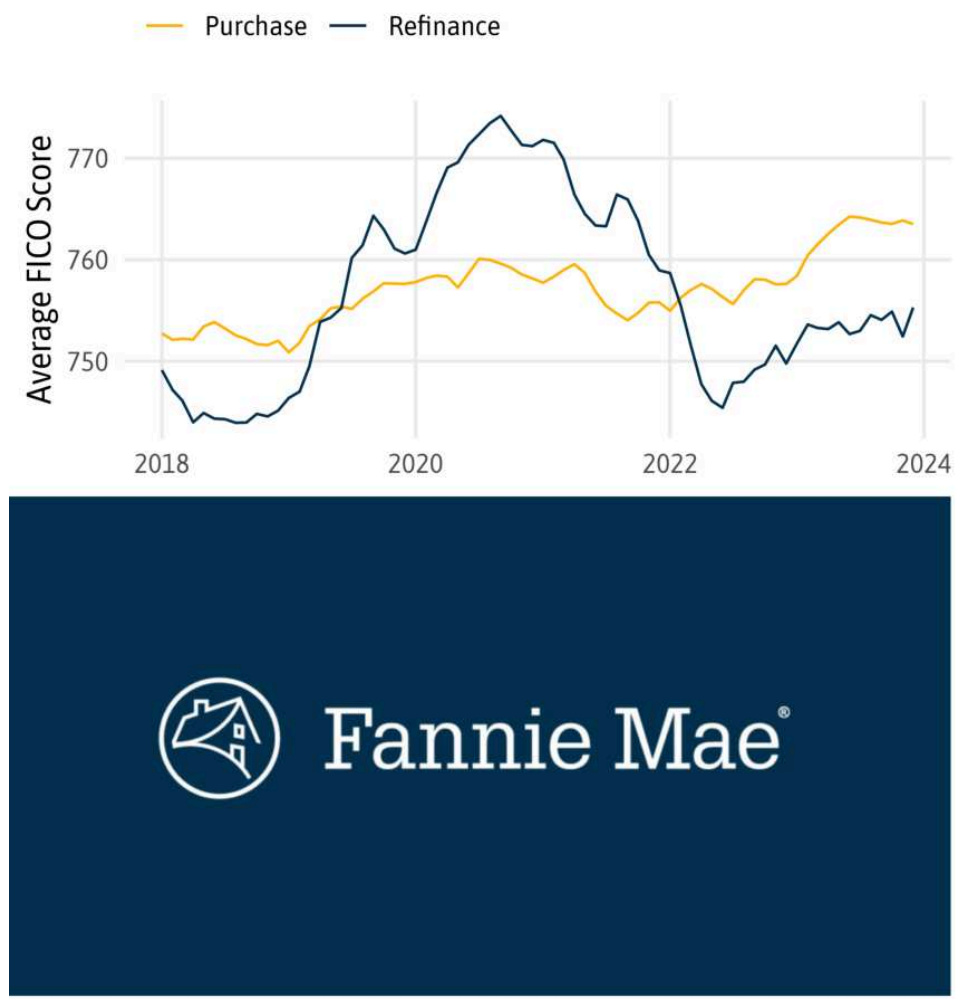
... and even lists:

1. this
2. is
3. awesome!



Insert Images with {marquee}

```
logo <- "![](img/logo.jpg)"

g +
  labs(caption = logo) +
  theme(
    plot.caption = marquee::element
      width = unit(1.15, "npc")
    )
  )
```



Insert Images with {marquee}

```
logo <- "Data by ![](img/logo-wide.

g +
  labs(caption = logo) +
  theme(
    plot.caption = marquee::element
      size = 20, margin = margin(20
    )
  )
```

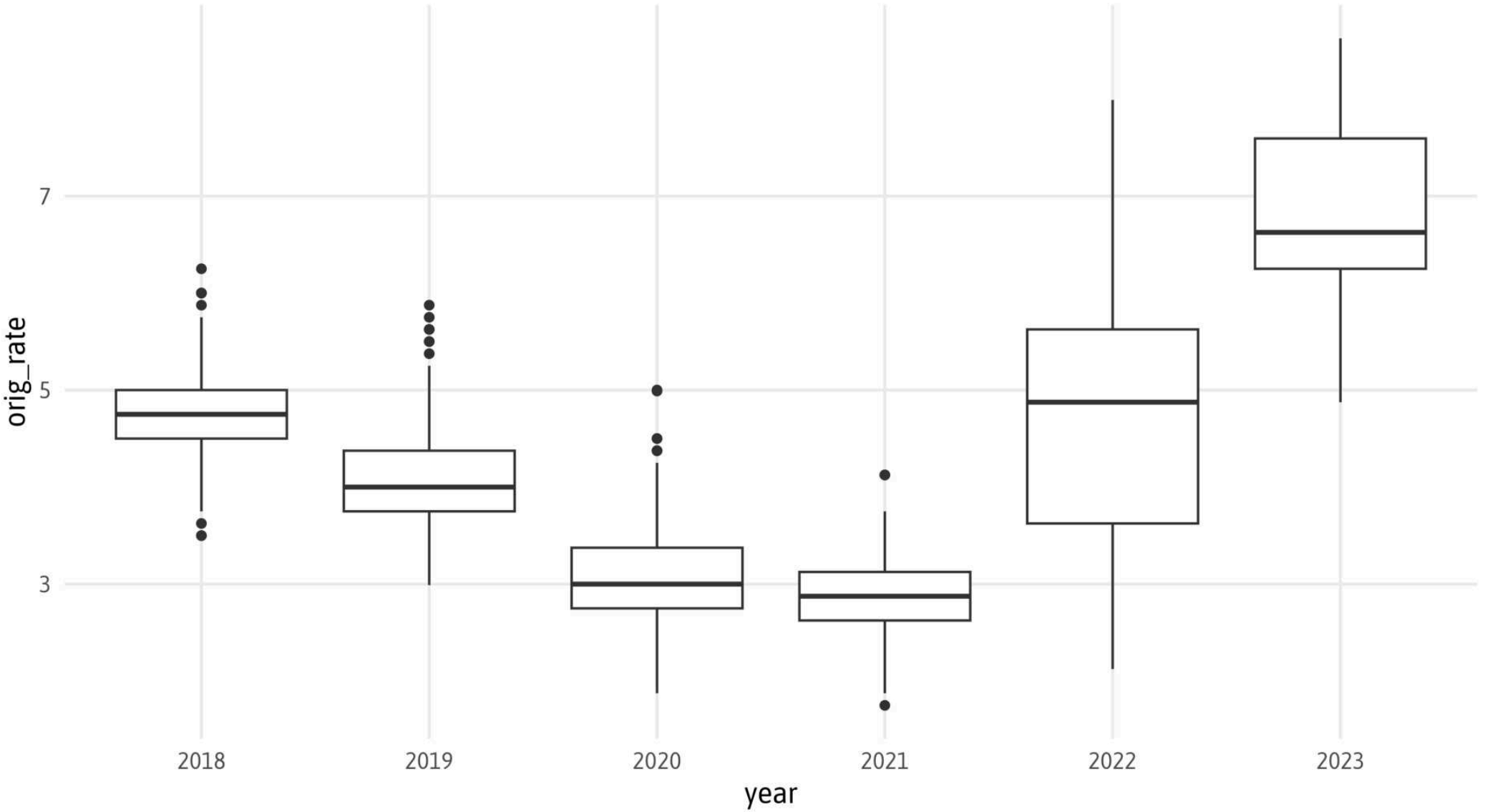
Insert Images with {marquee}

```
logo <- "*![](img/logo-wide.png)"

g +
  labs(caption = logo) +
  theme(
    plot.caption = marquee::element
      size = 20, margin = margin(20
      color = "white"
    )
  )
```

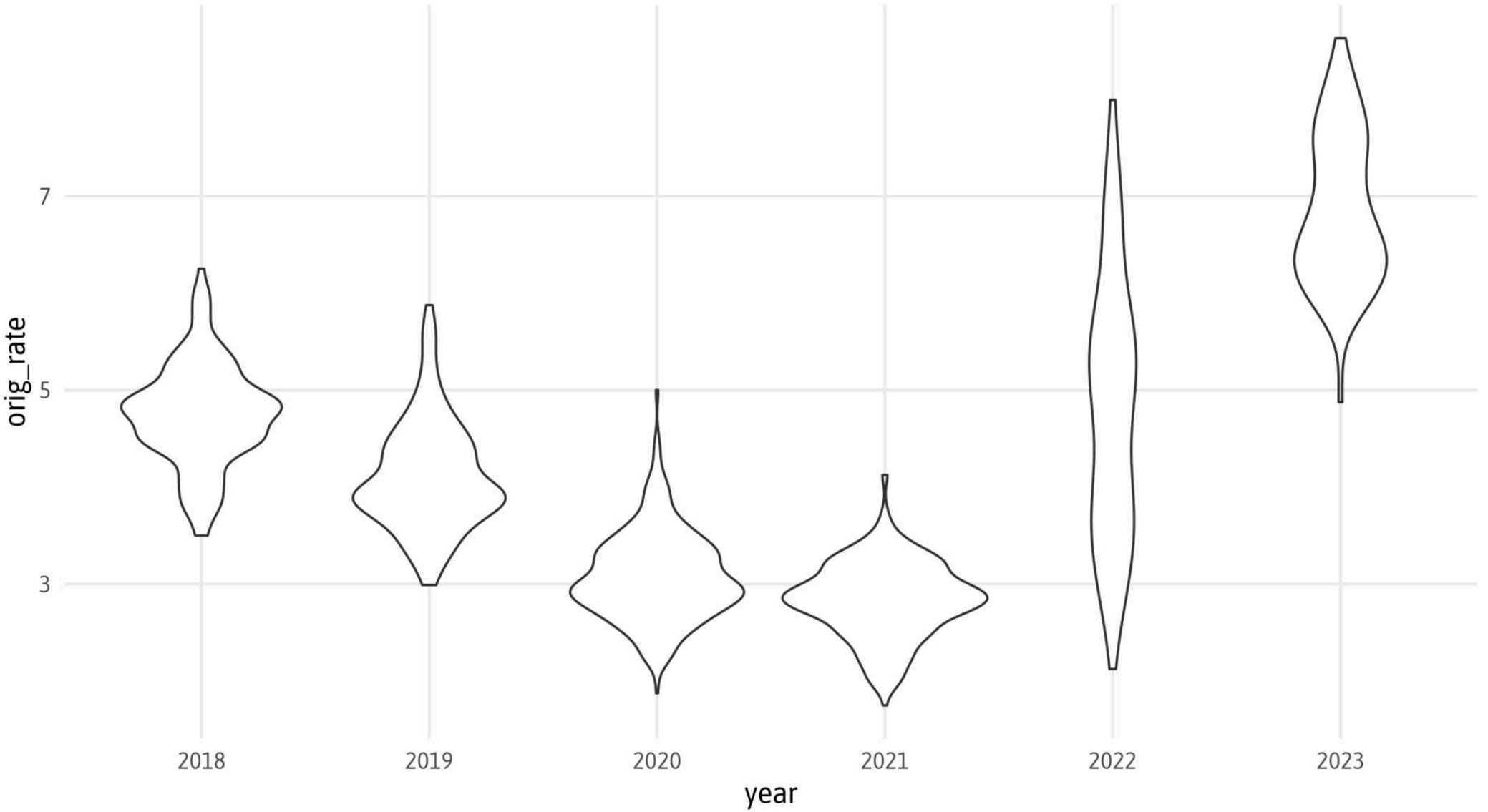
Visualizing Distributions

Cédric Scherer Data Visualization & Information Design

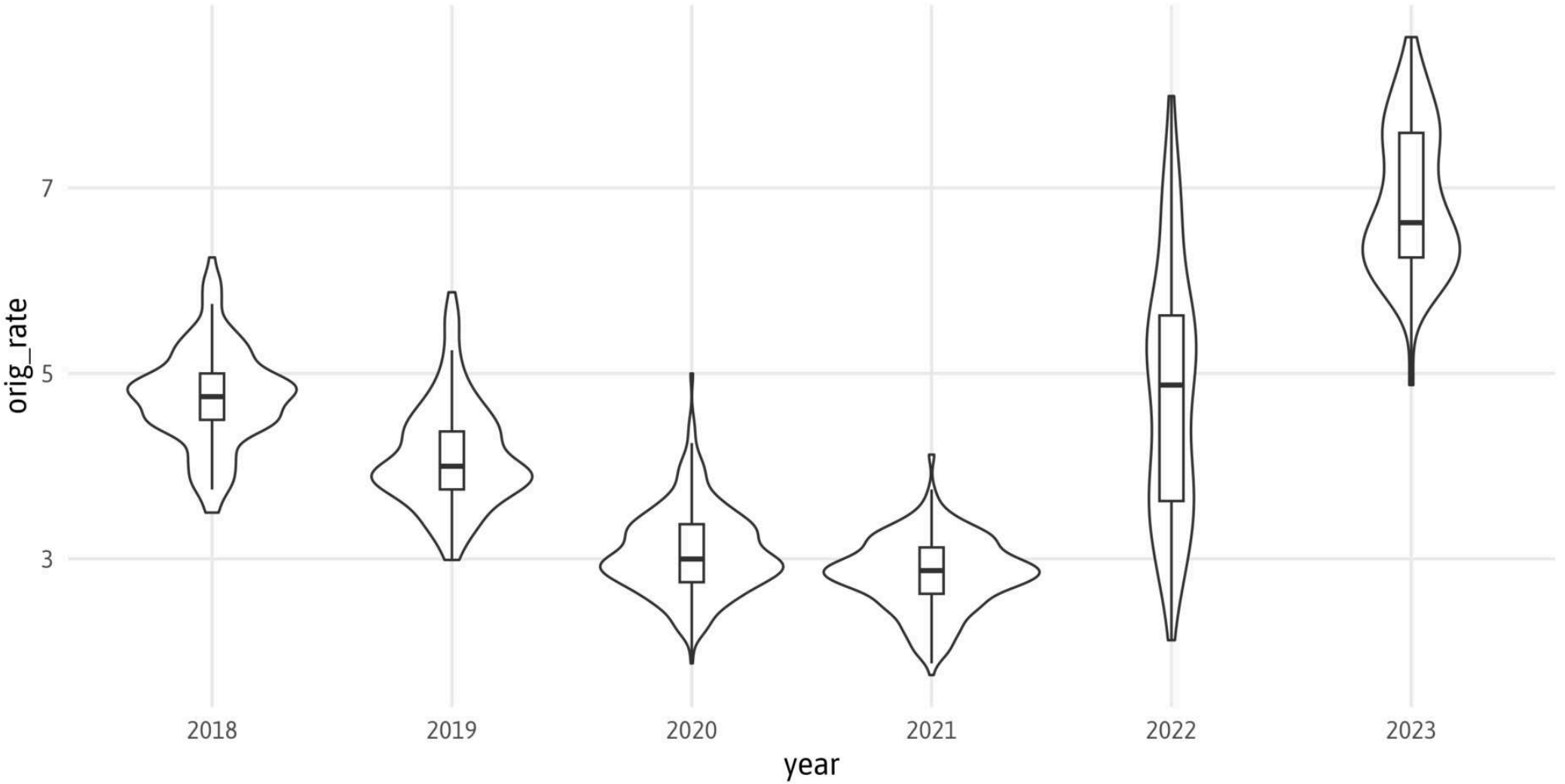
```
1 ggplot(filter(fannie_sub, first_flag == "N"), aes(x = year, y = orig_rate)) +
2   geom_boxplot()
```



```r
ggplot(filter(fannie_sub, first_flag == "N"), aes(x = year, y = orig_rate)) +
  geom_violin()
```



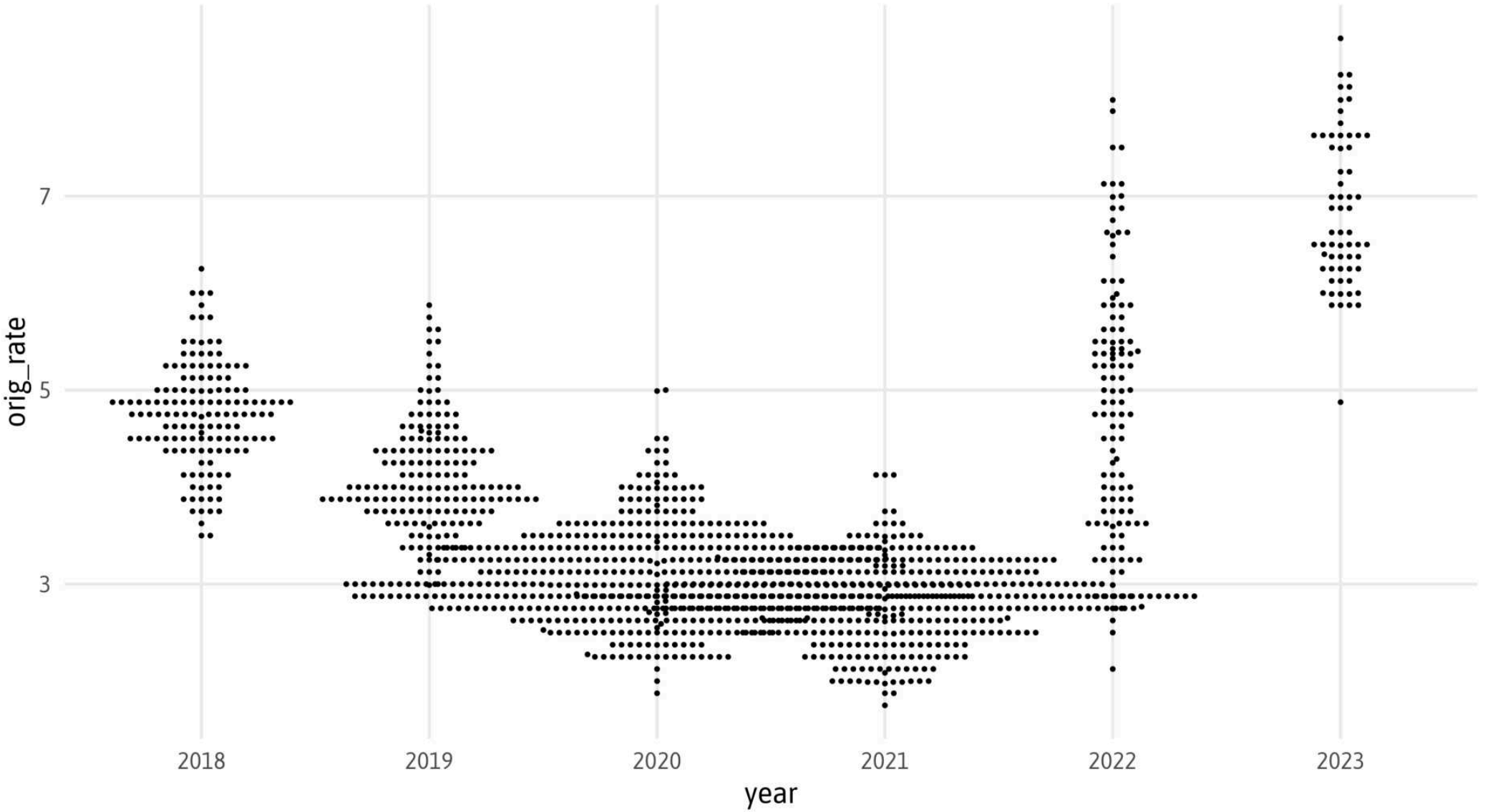
```r
ggplot(filter(fannie_sub, first_flag == "N"), aes(x = year, y = orig_rate)) +
  geom_violin() +
  geom_boxplot(width = .1, outlier.color = NA)
```



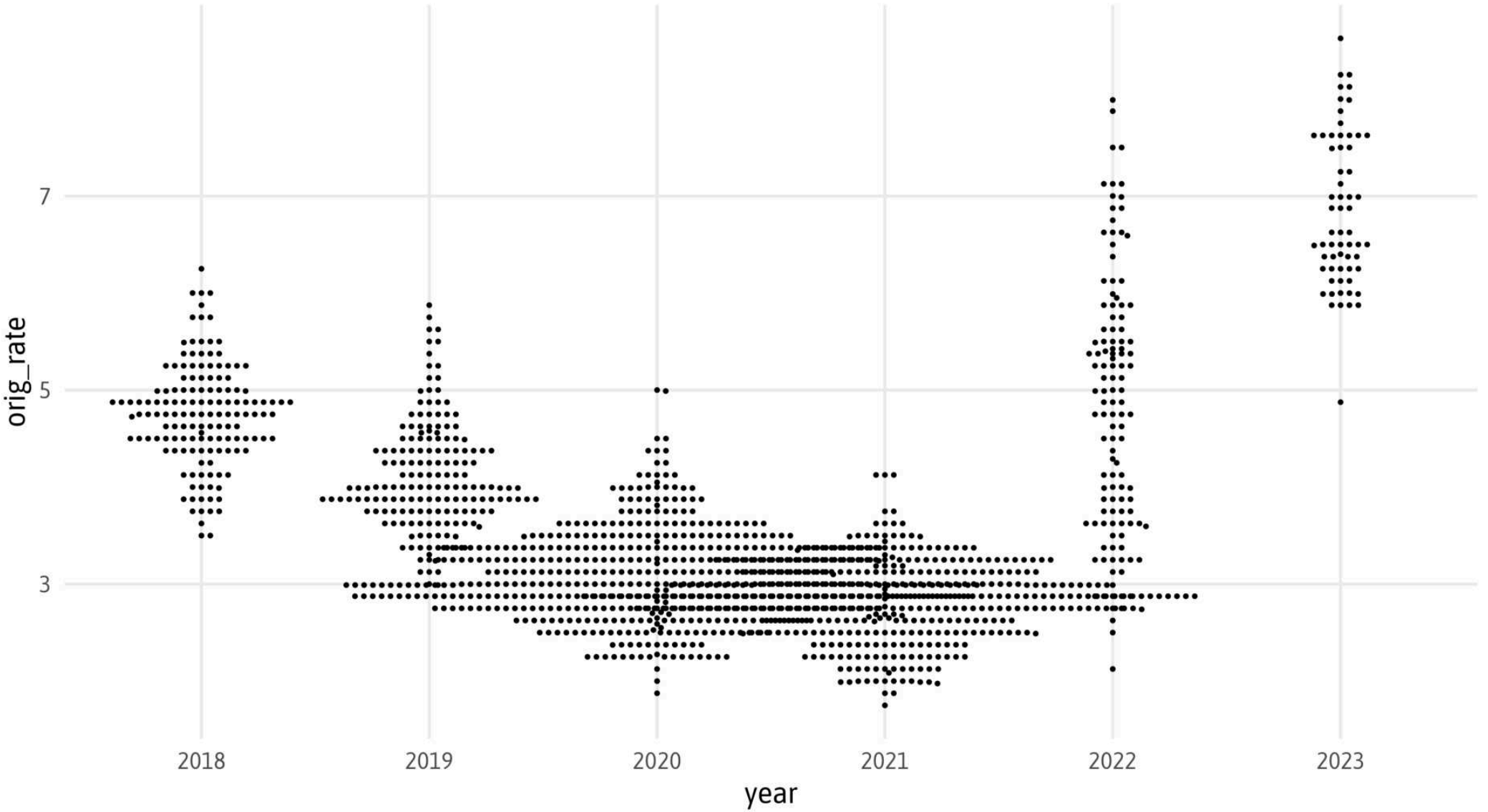
{ggbeeswarm}

Cédric Scherer Data Visualization & Information Design

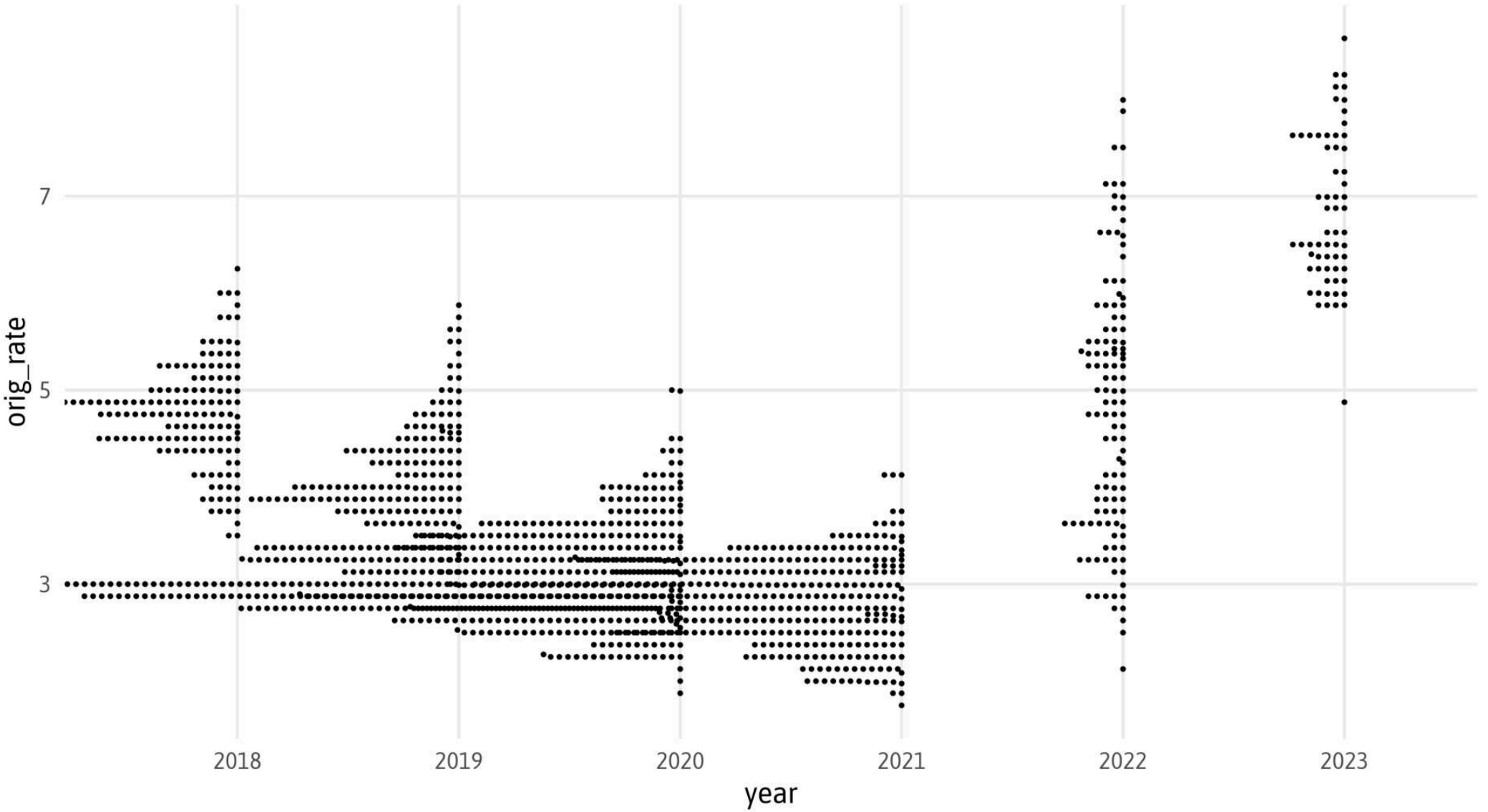
```
ggplot(filter(fannie_sub, first_flag == "N"), aes(x = year, y = orig_rate)) +
  ggbeeswarm::geom_beeswarm(cex = .65, size = .5)
```



```r
ggplot(filter(fannie_sub, first_flag == "N"), aes(x = year, y = orig_rate)) +
  ggbeeswarm::geom_beeswarm(cex = .65, size = .5, priority = "descending")
```

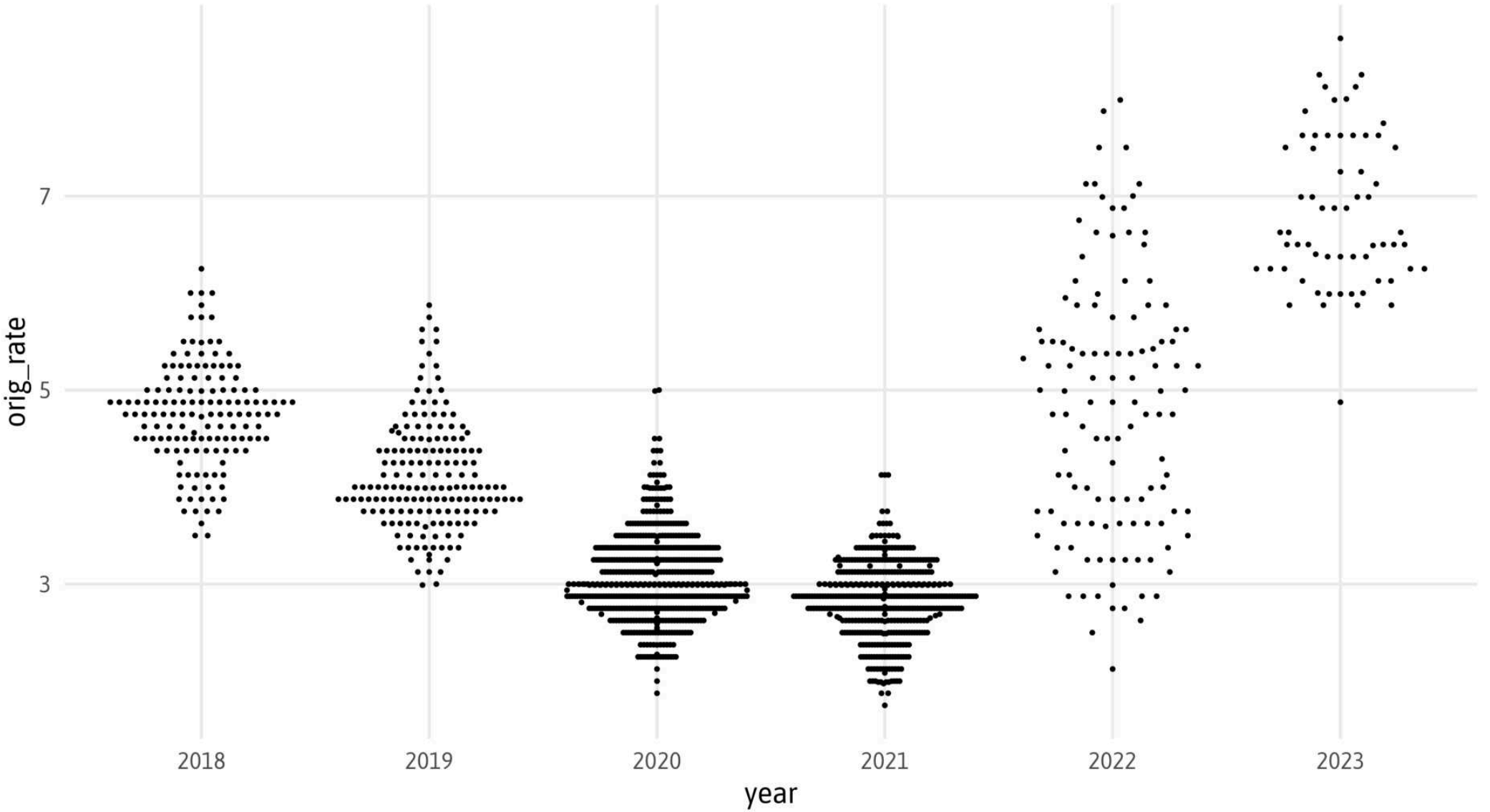


```r
ggplot(filter(fannie_sub, first_flag == "N"), aes(x = year, y = orig_rate)) +
  ggbeeswarm::geom_beeswarm(cex = .65, size = .5, side = -1)
```



```
ggplot(filter(fannie_sub, first_flag == "N"), aes(x = year, y = orig_rate)) +
  ggbeeswarm::geom_quasirandom(size = .5)
```

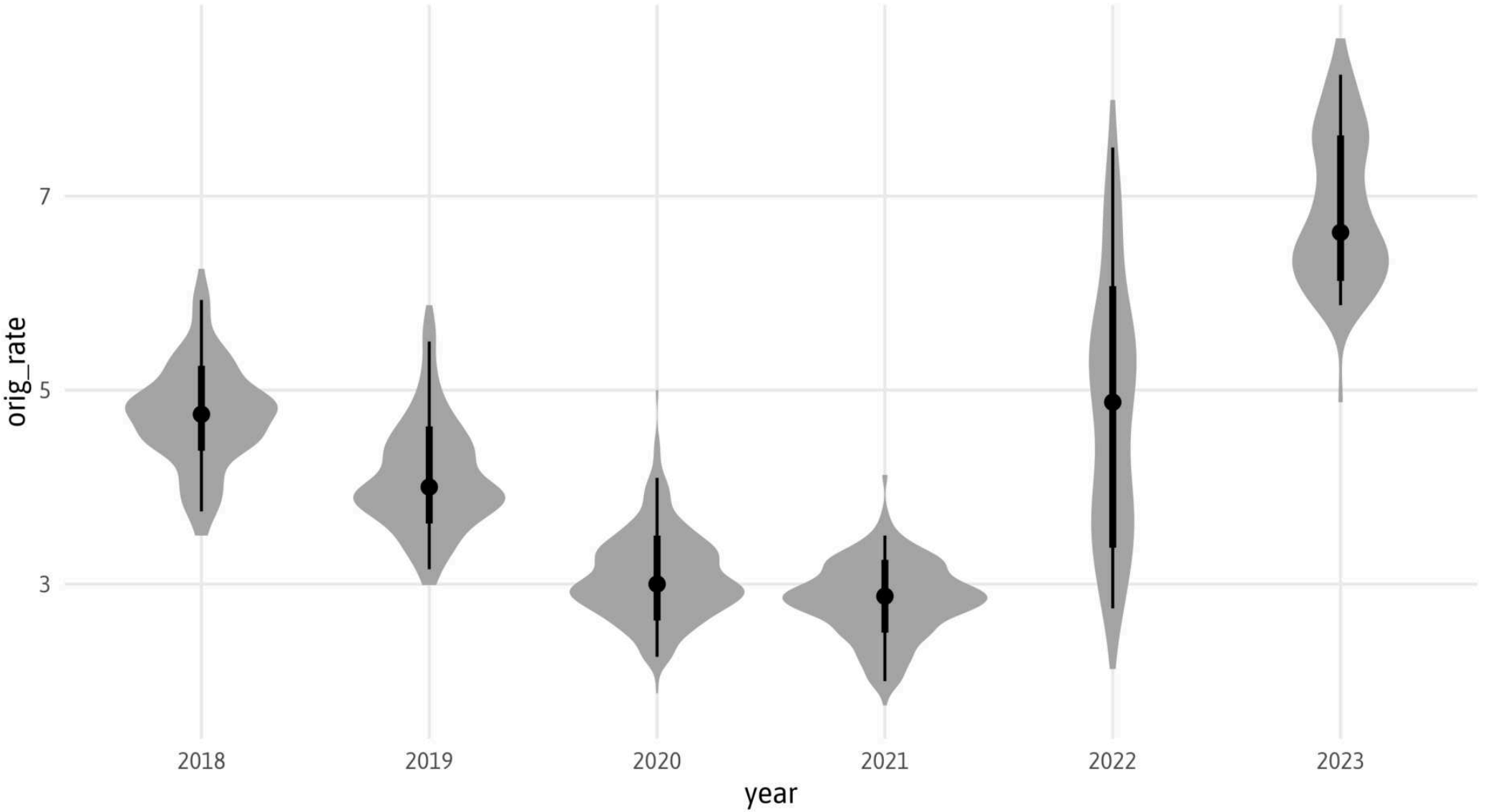
```
ggplot(filter(fannie_sub, first_flag == "N"), aes(x = year, y = orig_rate)) +
  ggbeeswarm::geom_quasirandom(size = .5, method = "smiley")
```



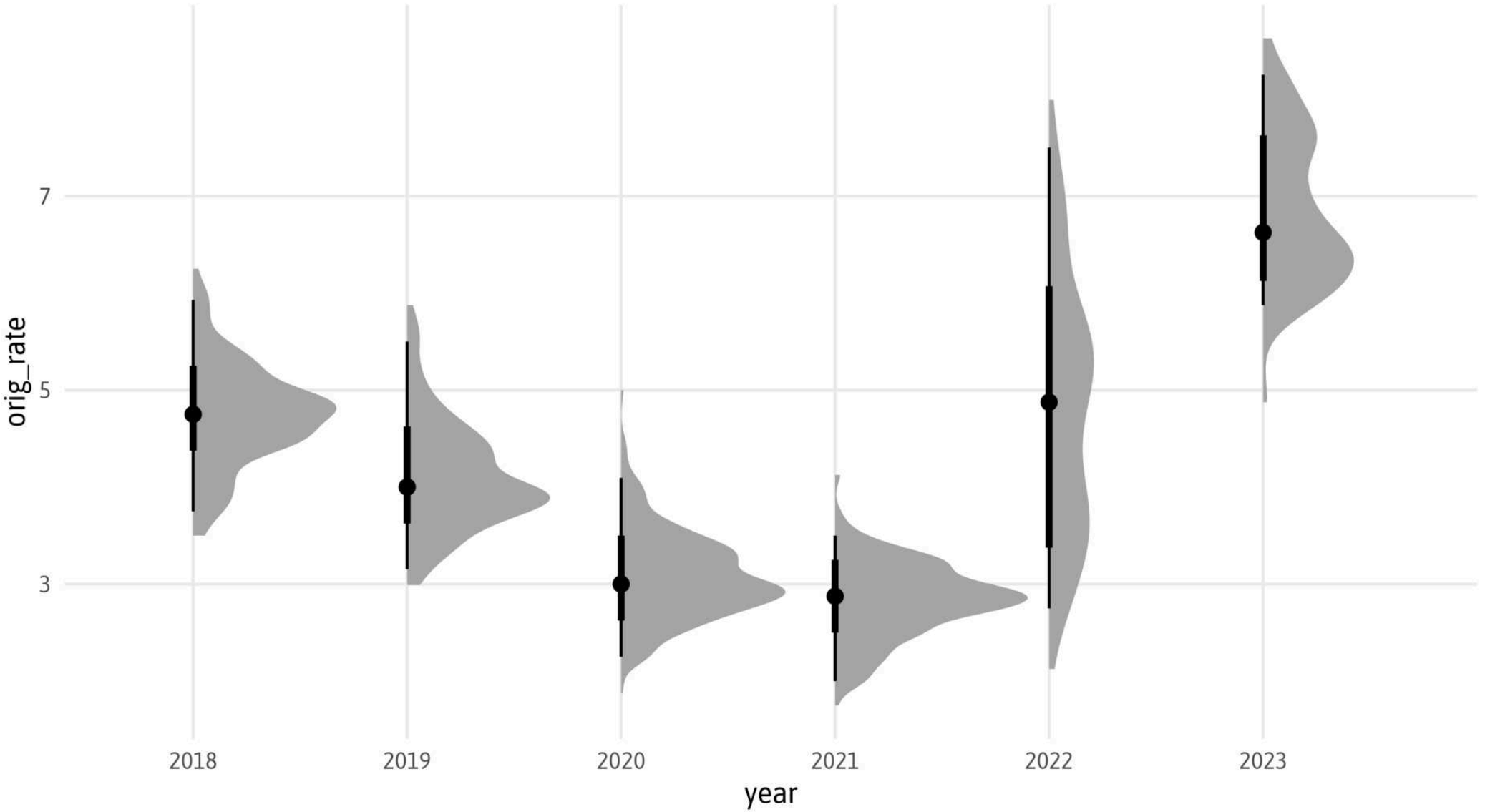
{ggdist}

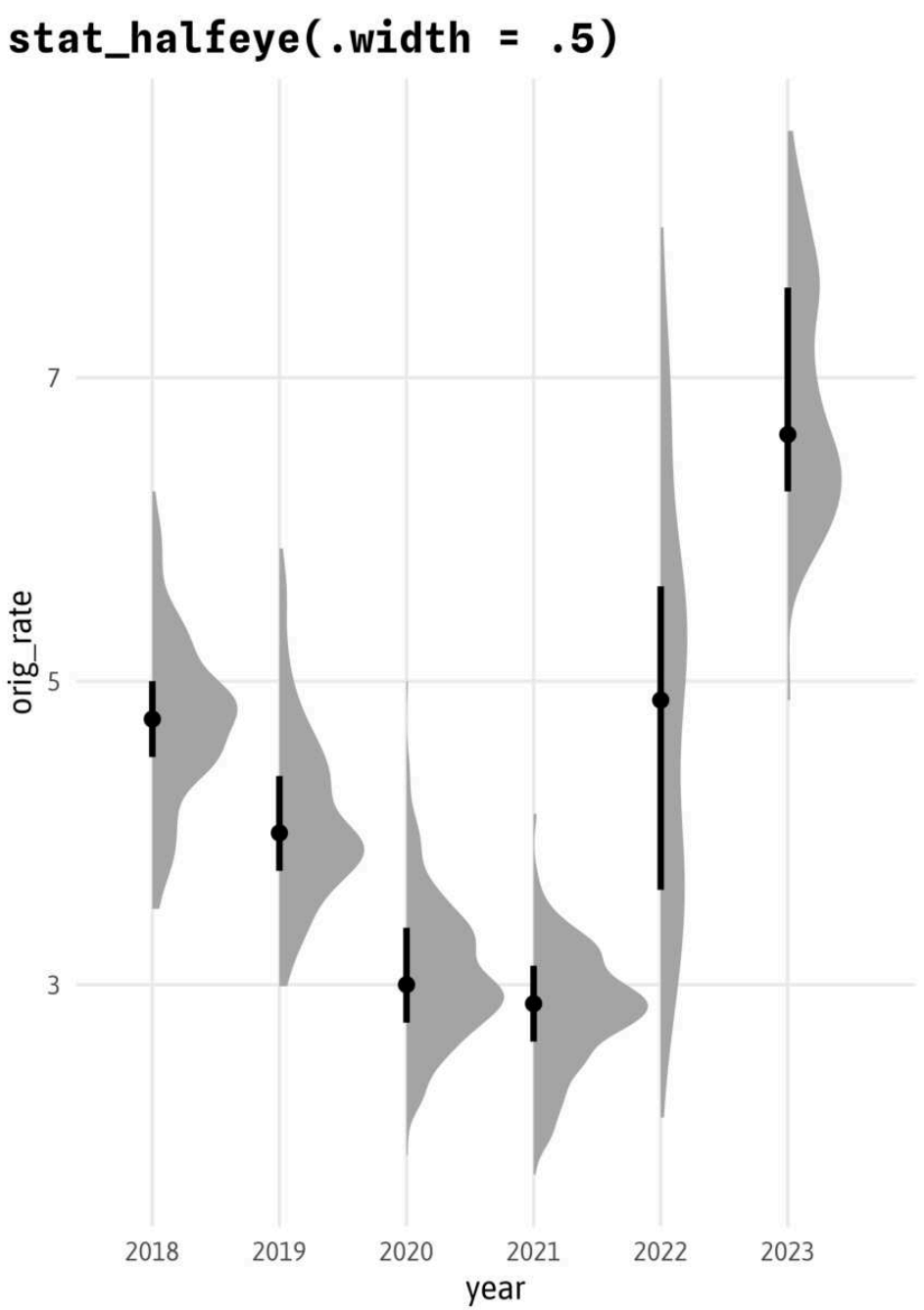
Cédric Scherer Data Visualization & Information Design

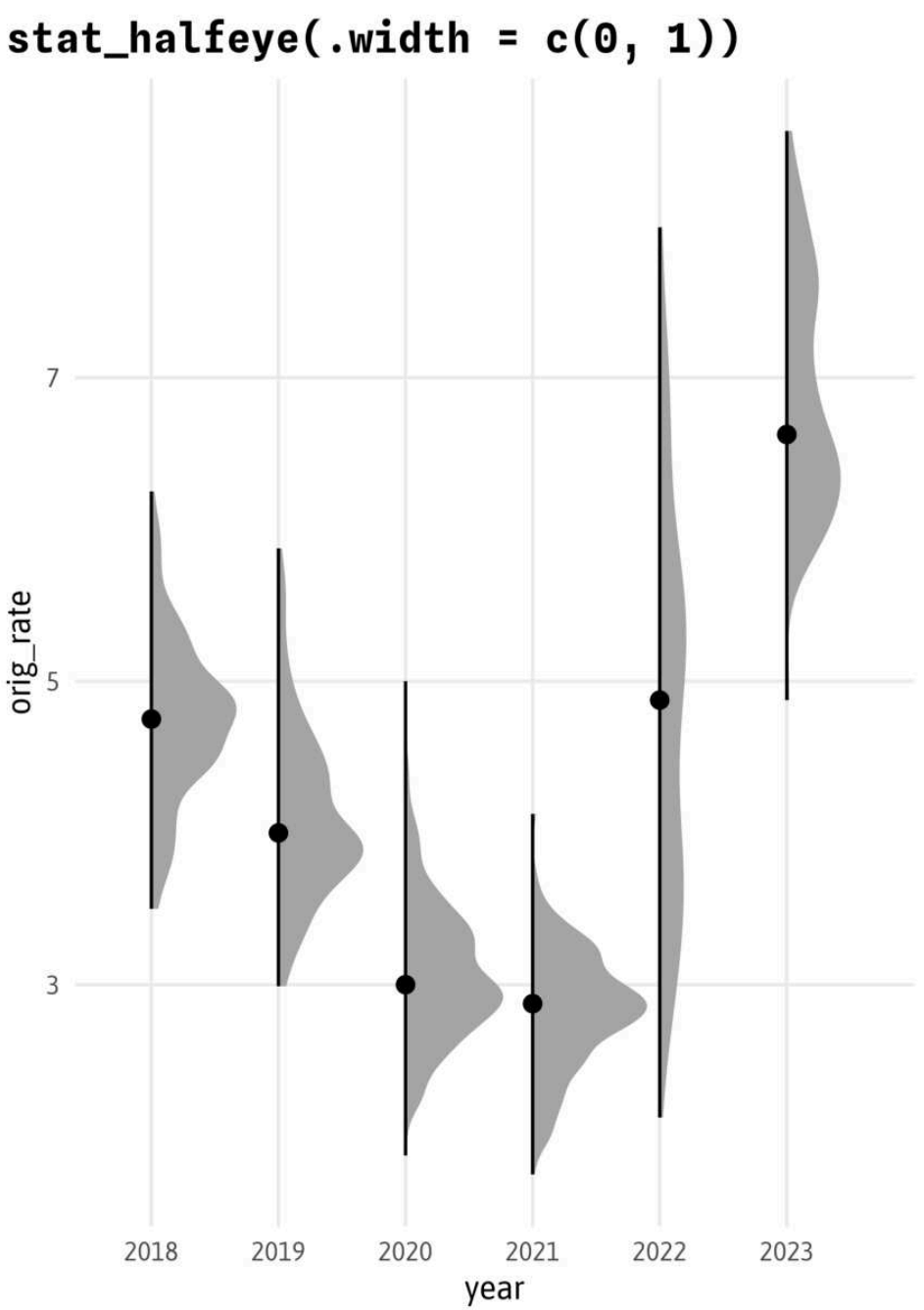
```r
ggplot(filter(fannie_sub, first_flag == "N"), aes(x = year, y = orig_rate)) +
  ggdist::stat_eye()
```



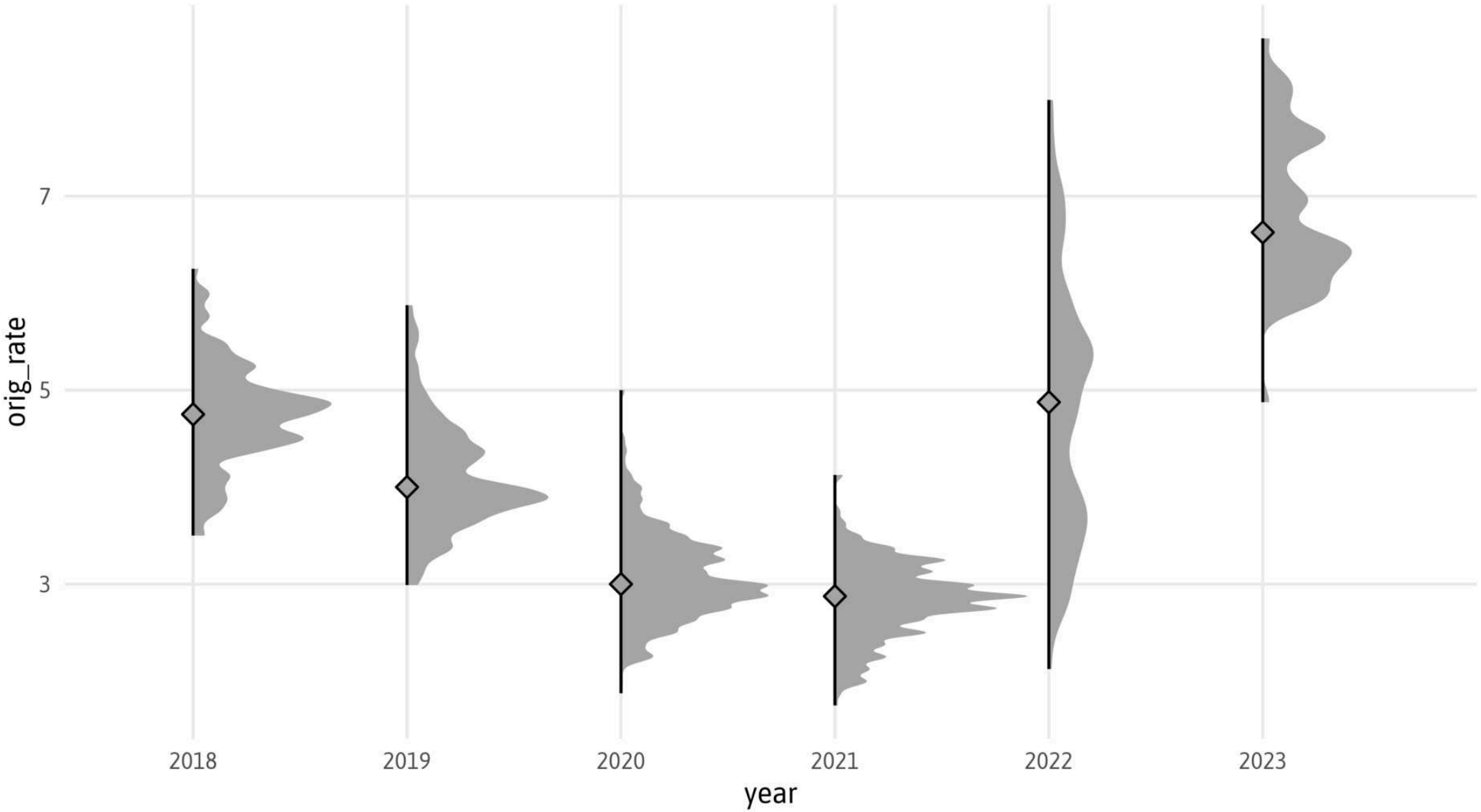
```
ggplot(filter(fannie_sub, first_flag == "N"), aes(x = year, y = orig_rate)) +
  ggdist::stat_halfeye() ## default: `.width = c(.66, .95)`
```

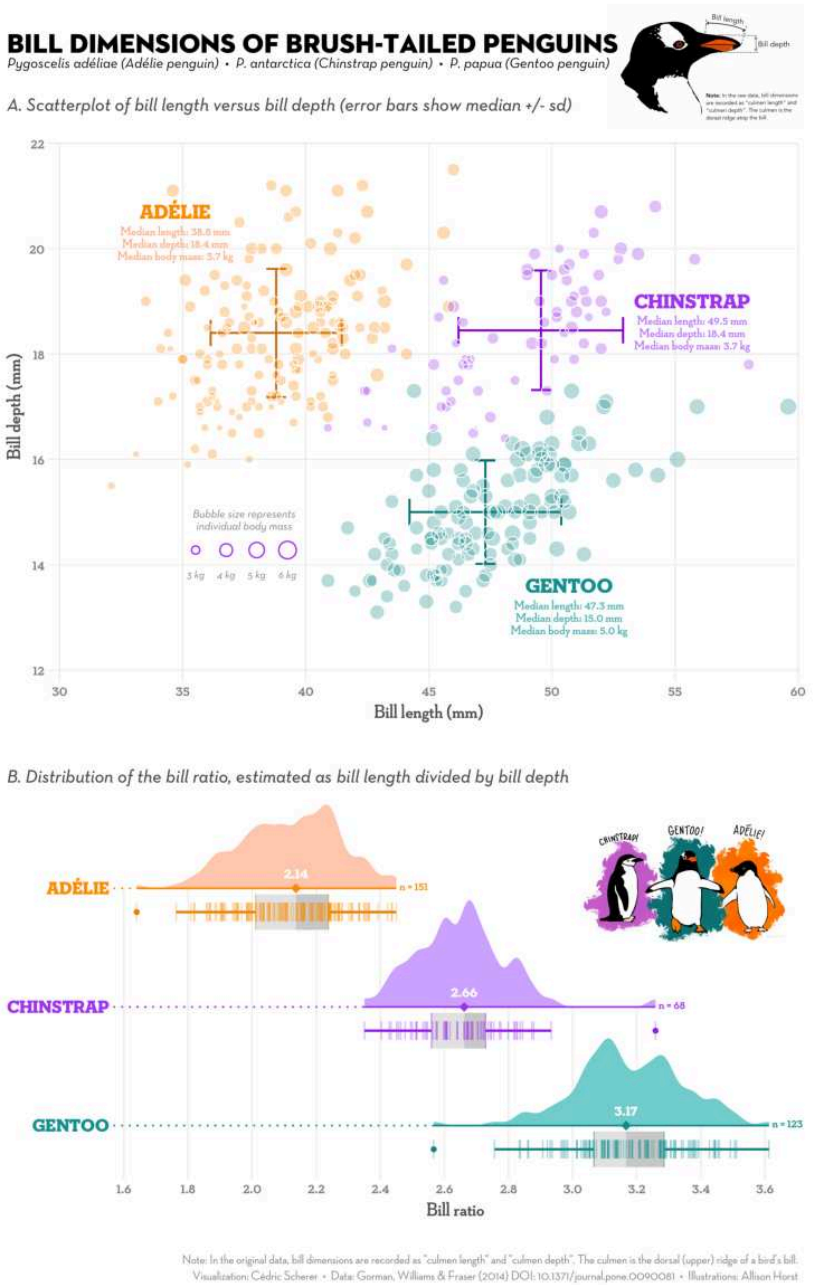



stat_halfeye(.width = .5)
orig_rate
7
5
3
2018
2019
2020
2021
2022
2023
year


stat_halfeye(.width = c(0, 1))
orig_rate
7
5
3
2018
2019
2020
2021
2022
2023
year

```r
ggplot(filter(fannie_sub, first_flag == "N"), aes(x = year, y = orig_rate)) +
  ggdist::stat_halfeye(.width = c(0, 1), adjust = .5, shape = 23, point_size = 3)
```



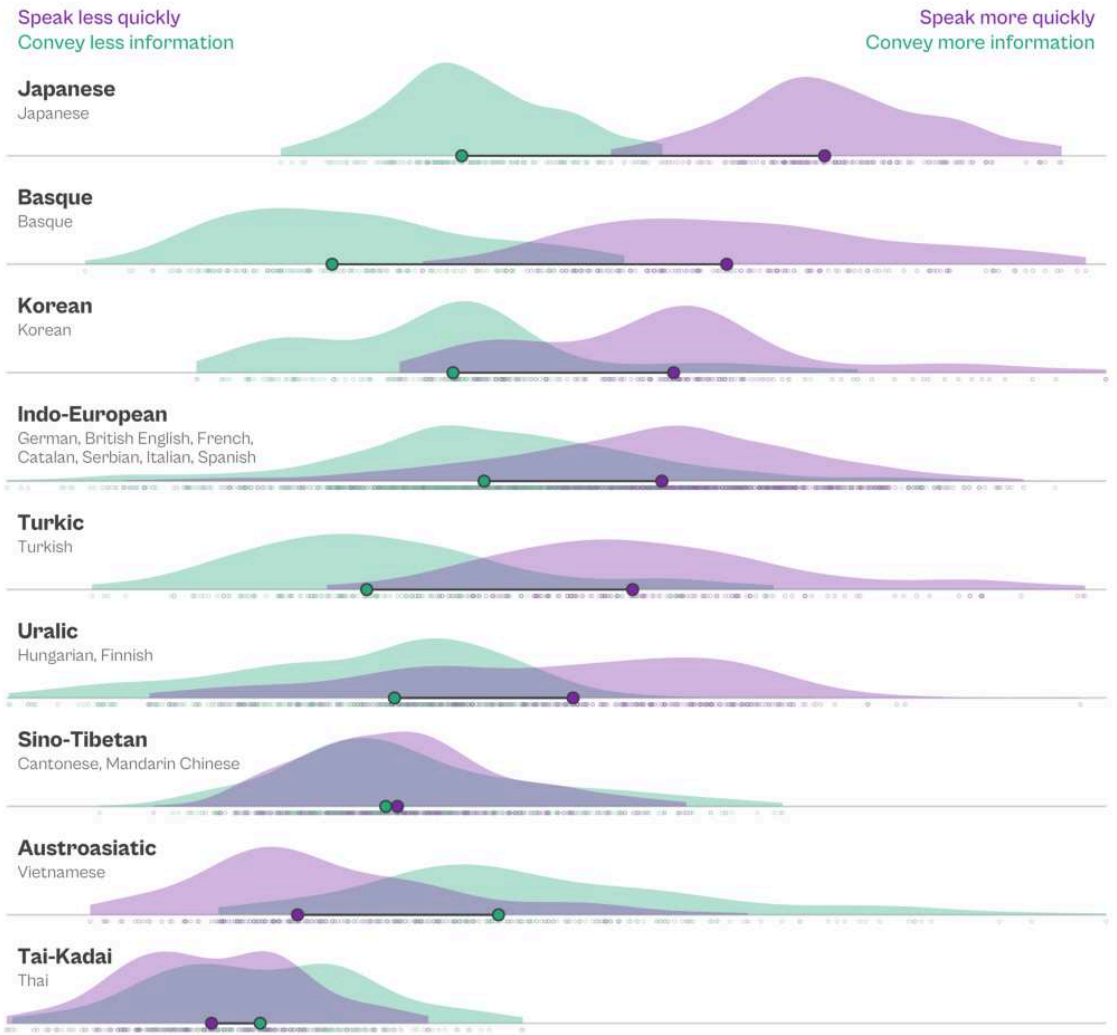


"Bill Dimensions of Brush-Tailed Penguins"

Cédric Scherer Data Visualization & Information Design

Communicating fast doesn't necessarily mean communicating more

Variation in speech and information rates across language families, shown as normalized rates for comparison. While there are stark cross-linguistic differences in speech rates, information rates are more similar.



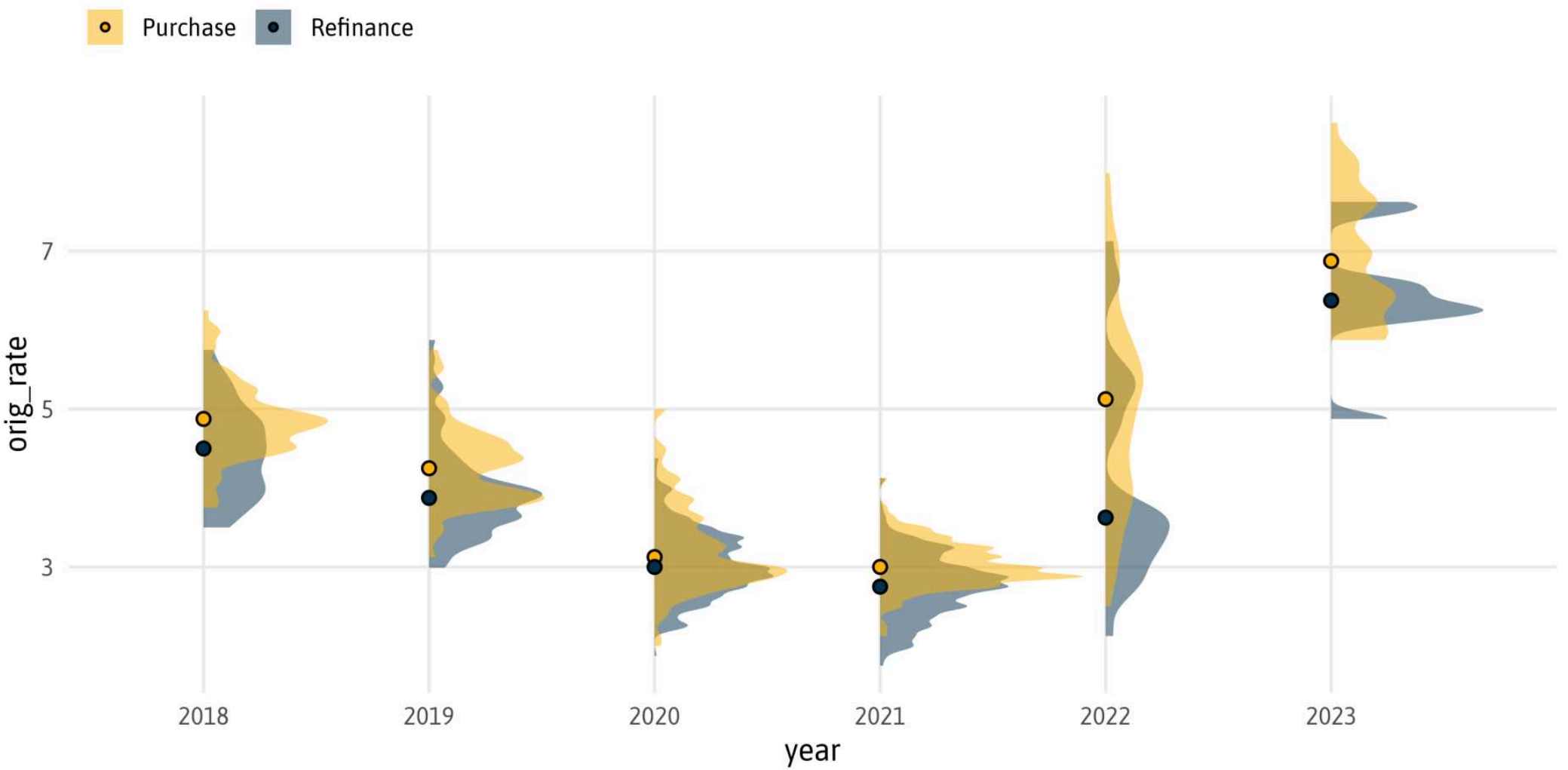
Normalized rates of speech (syllables per second) and information (bits per second)

Source: Coupé *et al.* 2019 *Science Advances* **5**(9). DOI: 10.1126/sciadv.aaw2594
Graphic: Cédric Scherer • Large dots show the median rates for each language family. Small dots show single estimates.

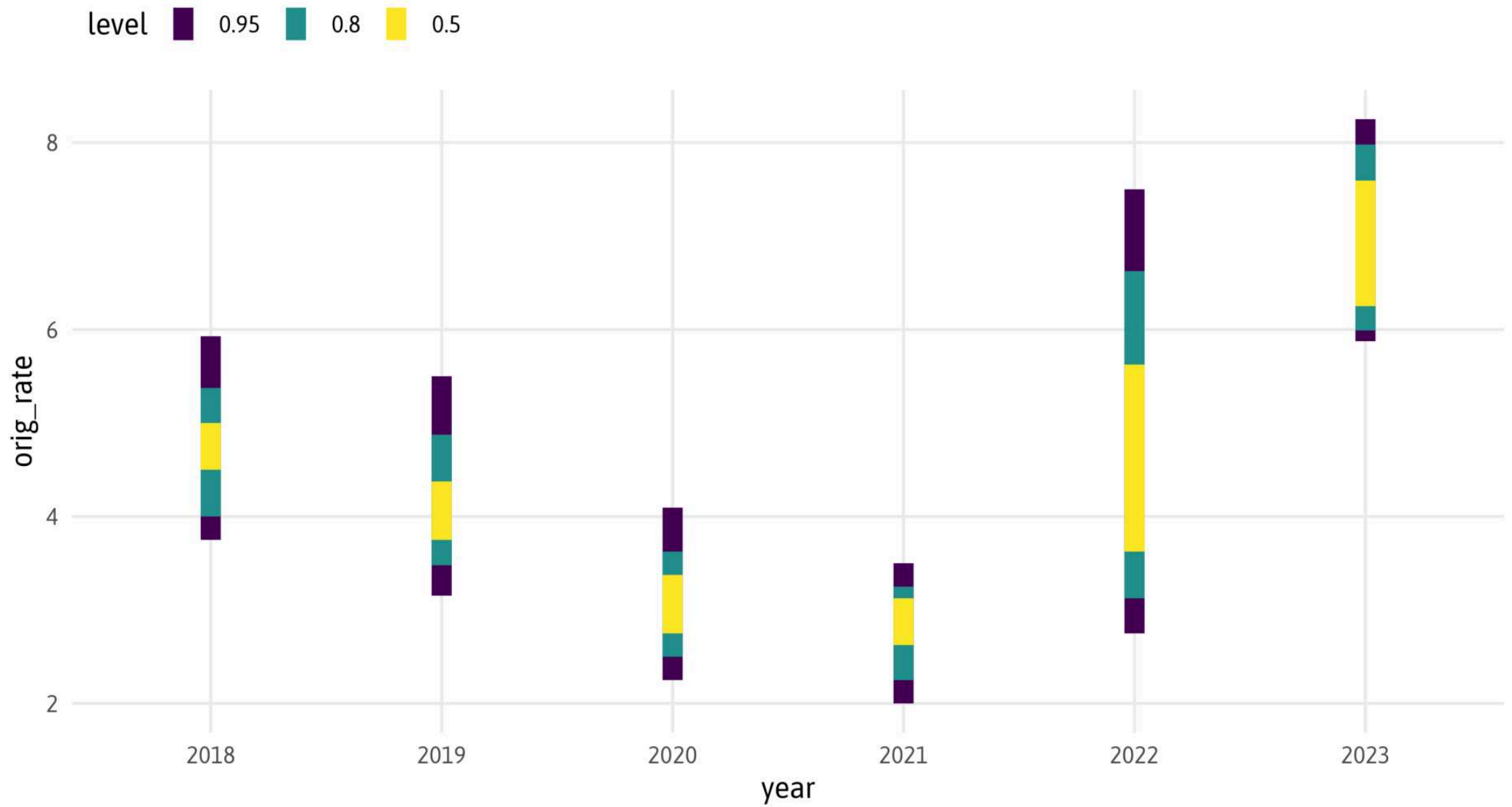
My reinterpreted The Economist graphic

Cédric Scherer Data Visualization & Information Design

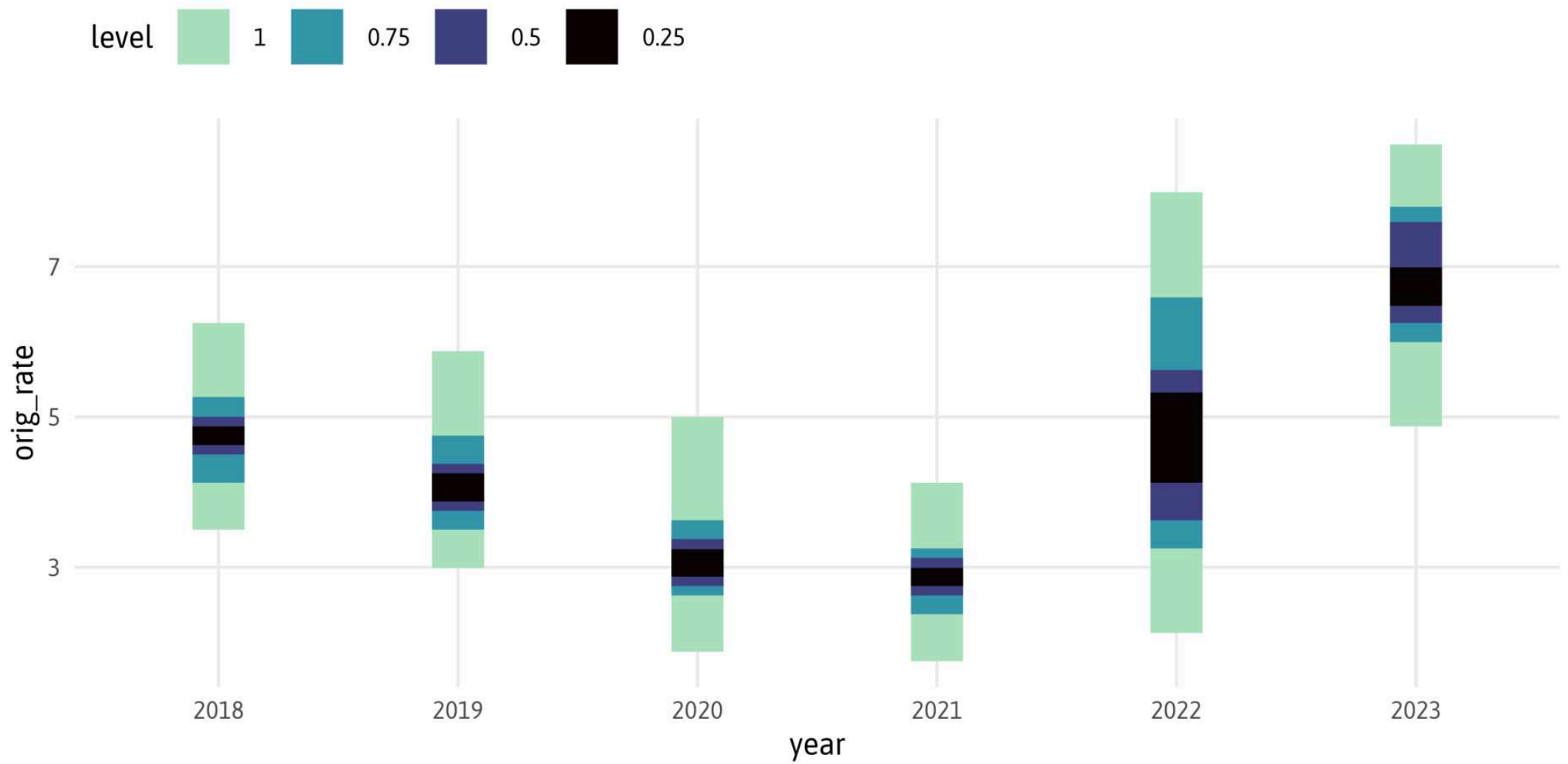
```r
ggplot(filter(fannie_sub, first_flag == "N"), aes(x = year, y = orig_rate, fill = purp
  ggdist::stat_halfeye(.width = 0, adjust = .5, slab_alpha = .5, shape = 21) +
  scale_fill_manual(values = c("#FFB306", "#05314D"), name = NULL)
```



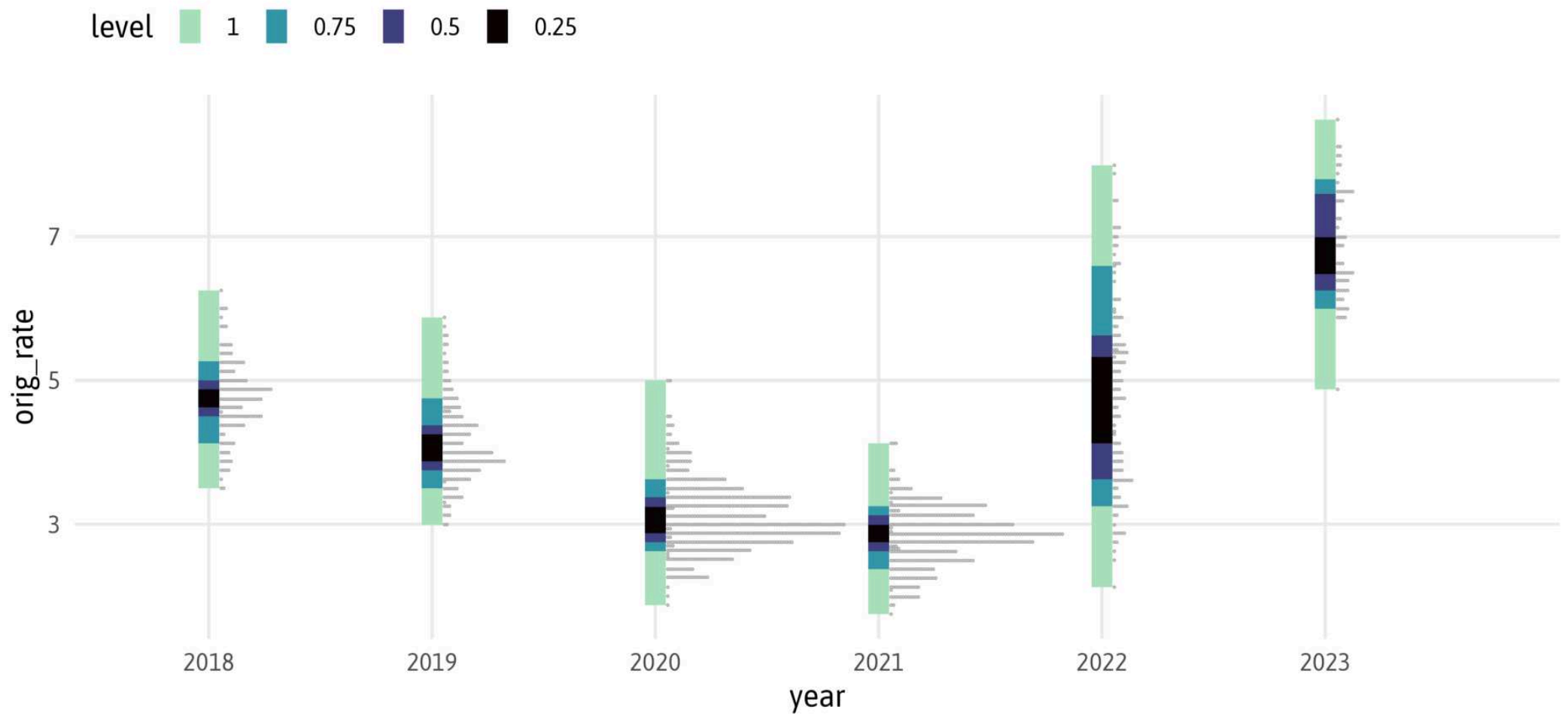
```
ggplot(filter(fannie_sub, first_flag == "N"), aes(x = year, y = orig_rate)) +
  ggdist::stat_interval()
```

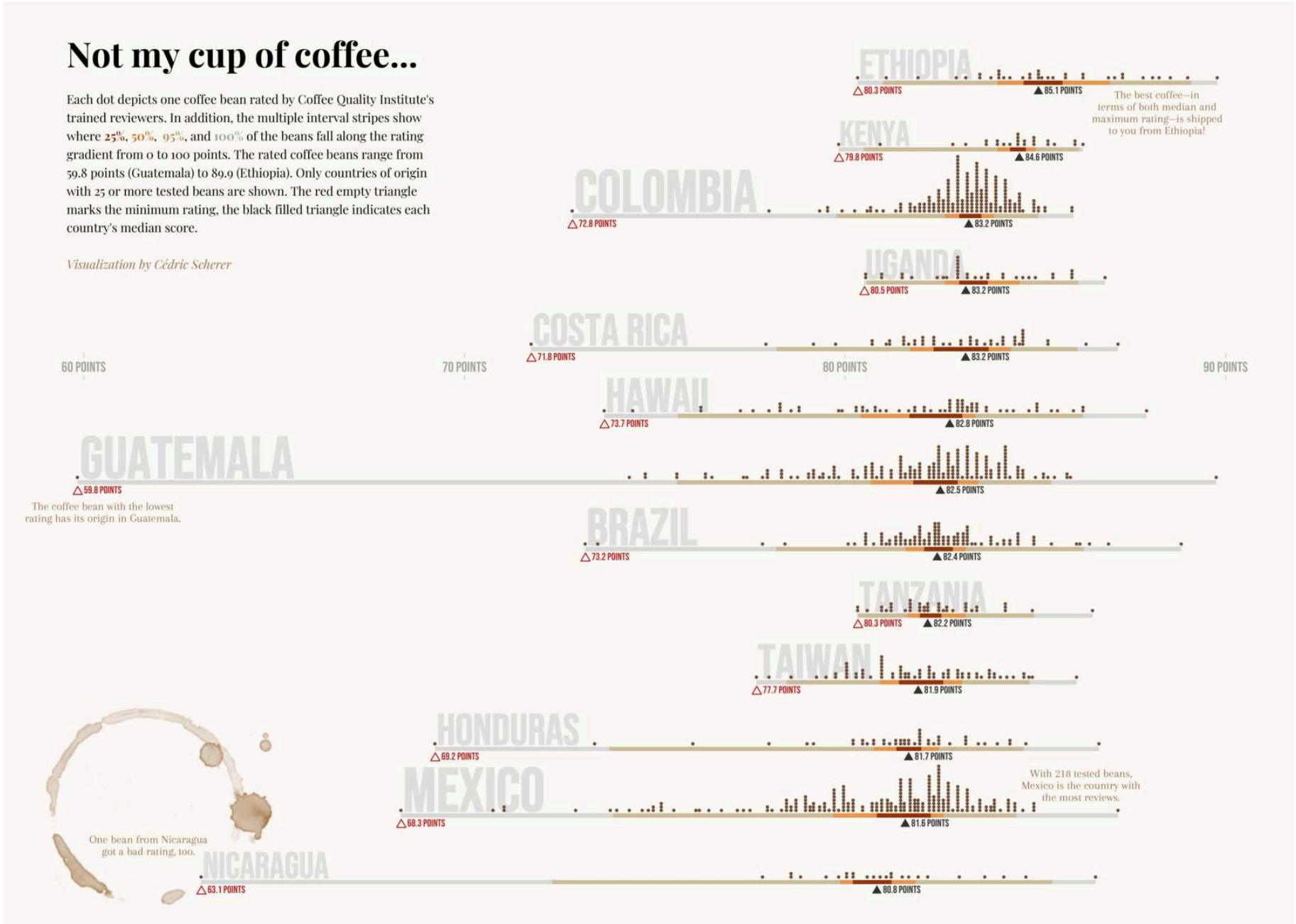


```
ggplot(filter(fannie_sub, first_flag == "N"), aes(x = year, y = orig_rate)) +
  ggdist::stat_interval(.width = 1:4*.25, linewidth = 10) +
  scale_color_viridis_d(option = "mako", direction = -1, end = .9)
```



```r
ggplot(filter(fannie_sub, first_flag == "N"), aes(x = year, y = orig_rate)) +
  ggdist::stat_interval(.width = 1:4*.25) +
  ggdist::stat_dots(position = position_nudge(x = .05), scale = .8) +
  scale_color_viridis_d(option = "mako", direction = -1, end = .9)
```

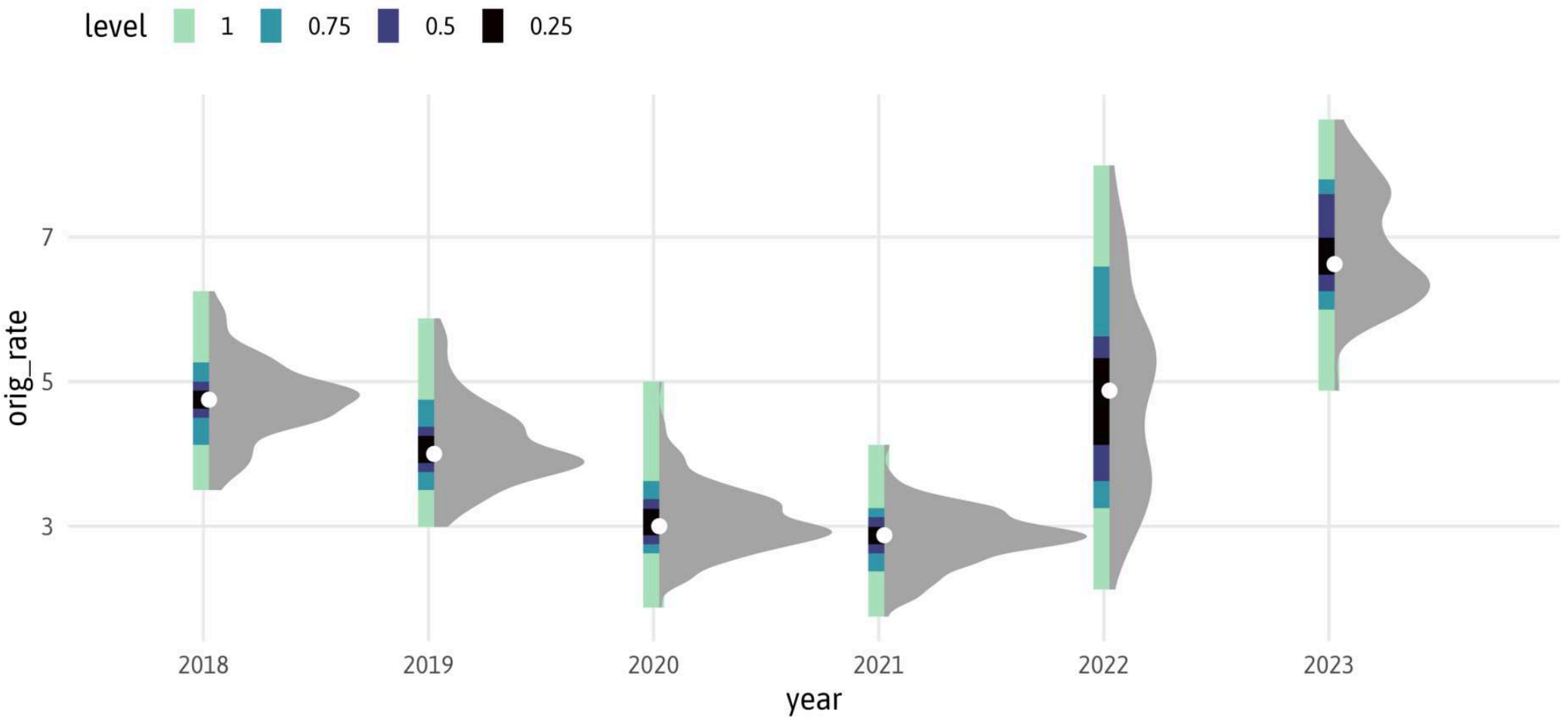




"Not My Cup of Coffee"

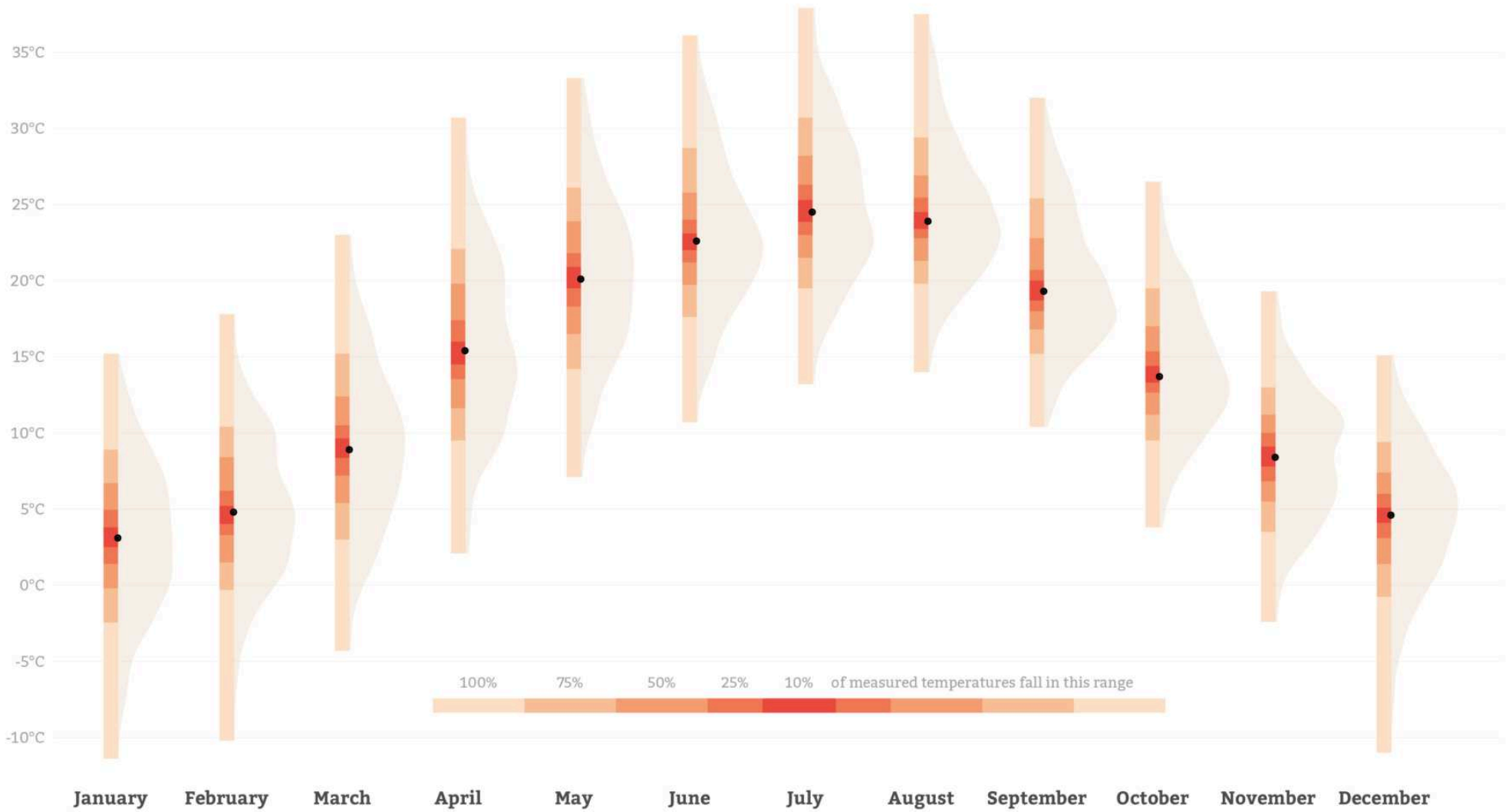
Cédric Scherer Data Visualization & Information Design

```r
ggplot(filter(fannie_sub, first_flag == "N"), aes(x = year, y = orig_rate)) +
  ggdist::stat_interval(.width = 1:4*.25) +
  ggdist::stat_halfeye(.width = 0, color = "white", position = position_nudge(x = .025
  scale_color_viridis_d(option = "mako", direction = -1, end = .9)
```



Daily Temperatures in Berlin, Germany

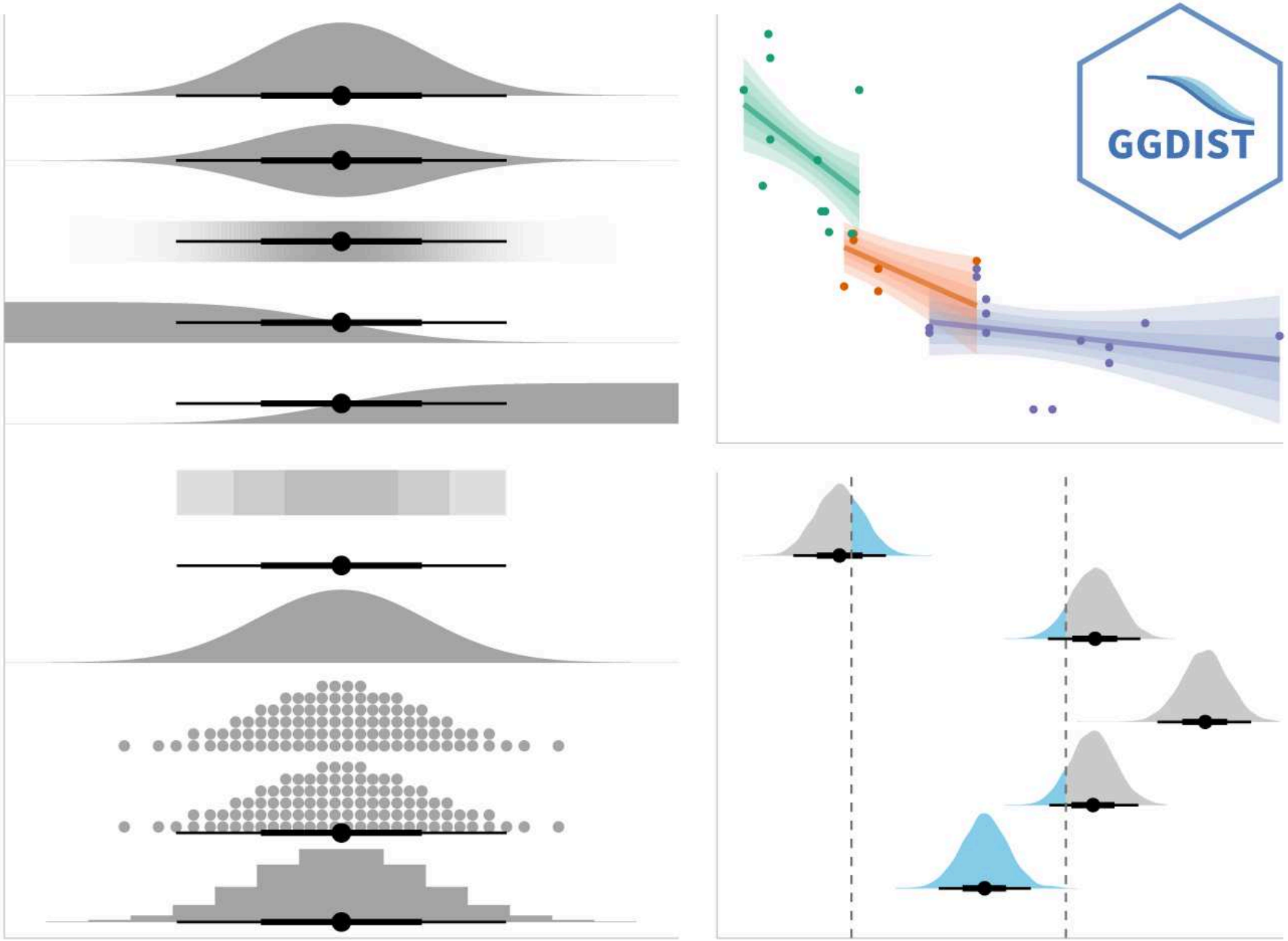
Range and distribution of maximum daily temperatures in Celsius per month from 2000 to 2018 measured in Berlin-Dahlem, Germany



Visualization by Cédric Scherer | Data: DWD (Deutscher Wetterdienst)

My Contribution to the SWD Challenge "Visualizing Uncertainty"

Cédric Scherer Data Visualization & Information Design

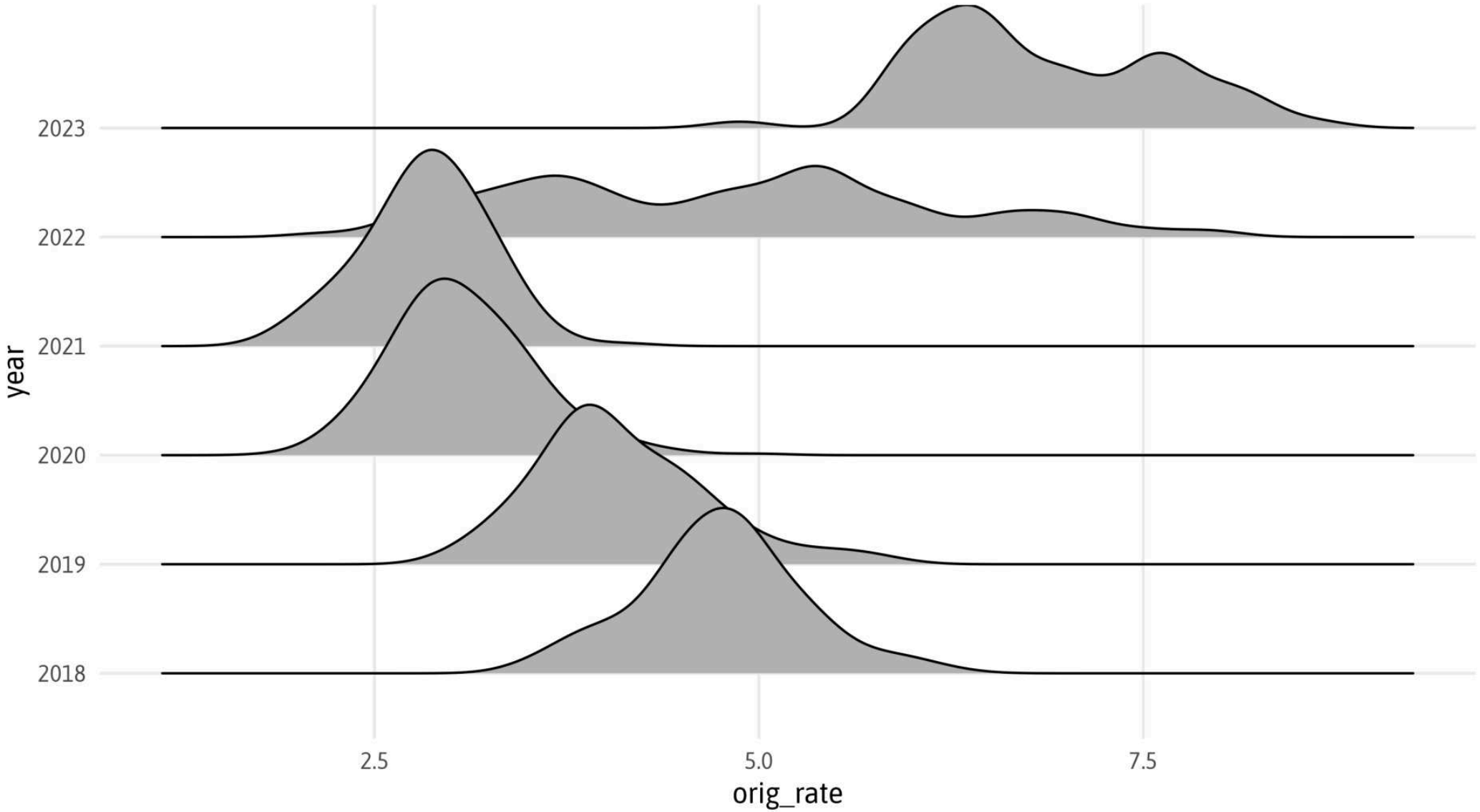


Source: mjskay.github.io/ggdist

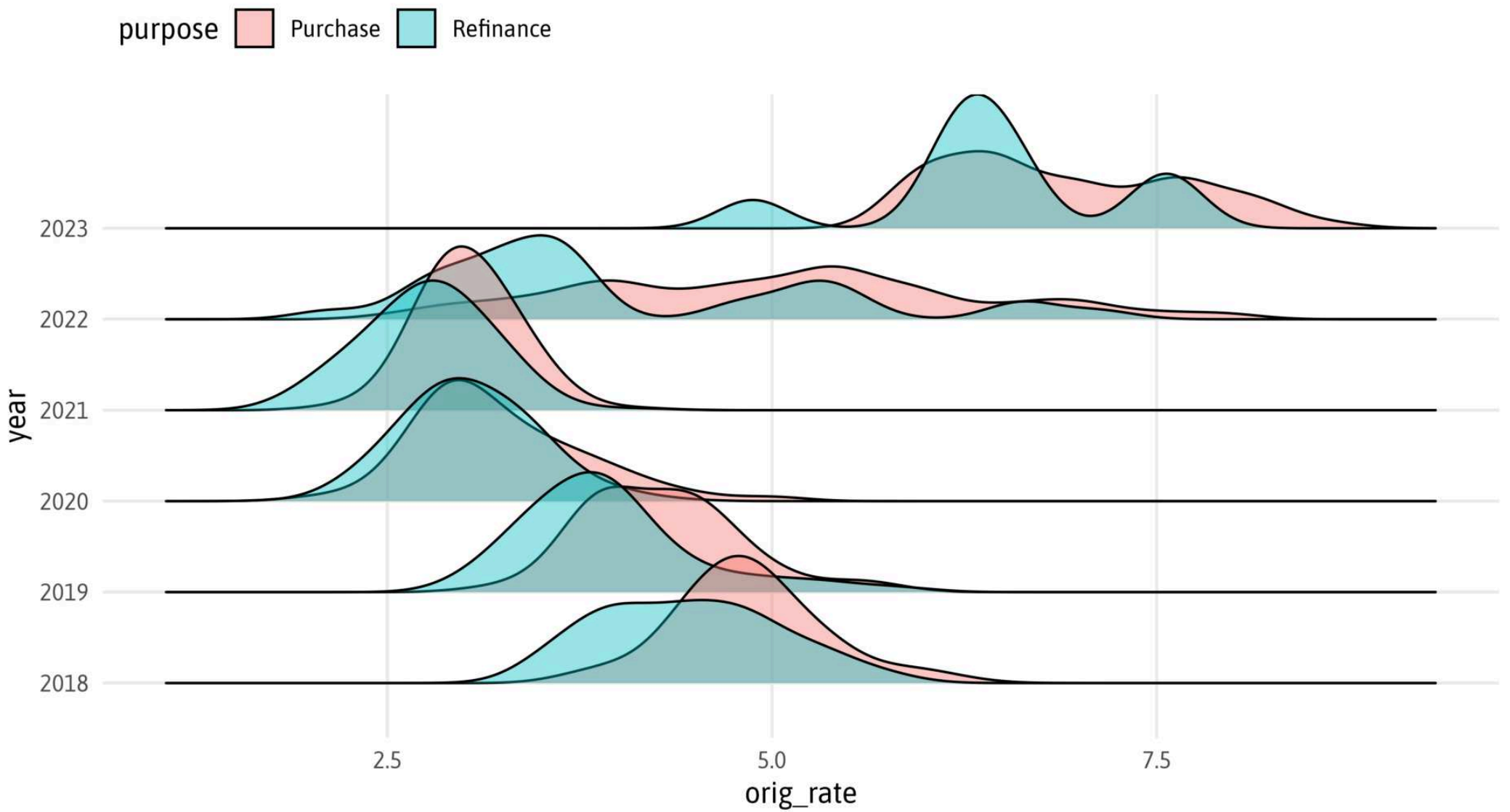
{ggridges}

Cédric Scherer Data Visualization & Information Design

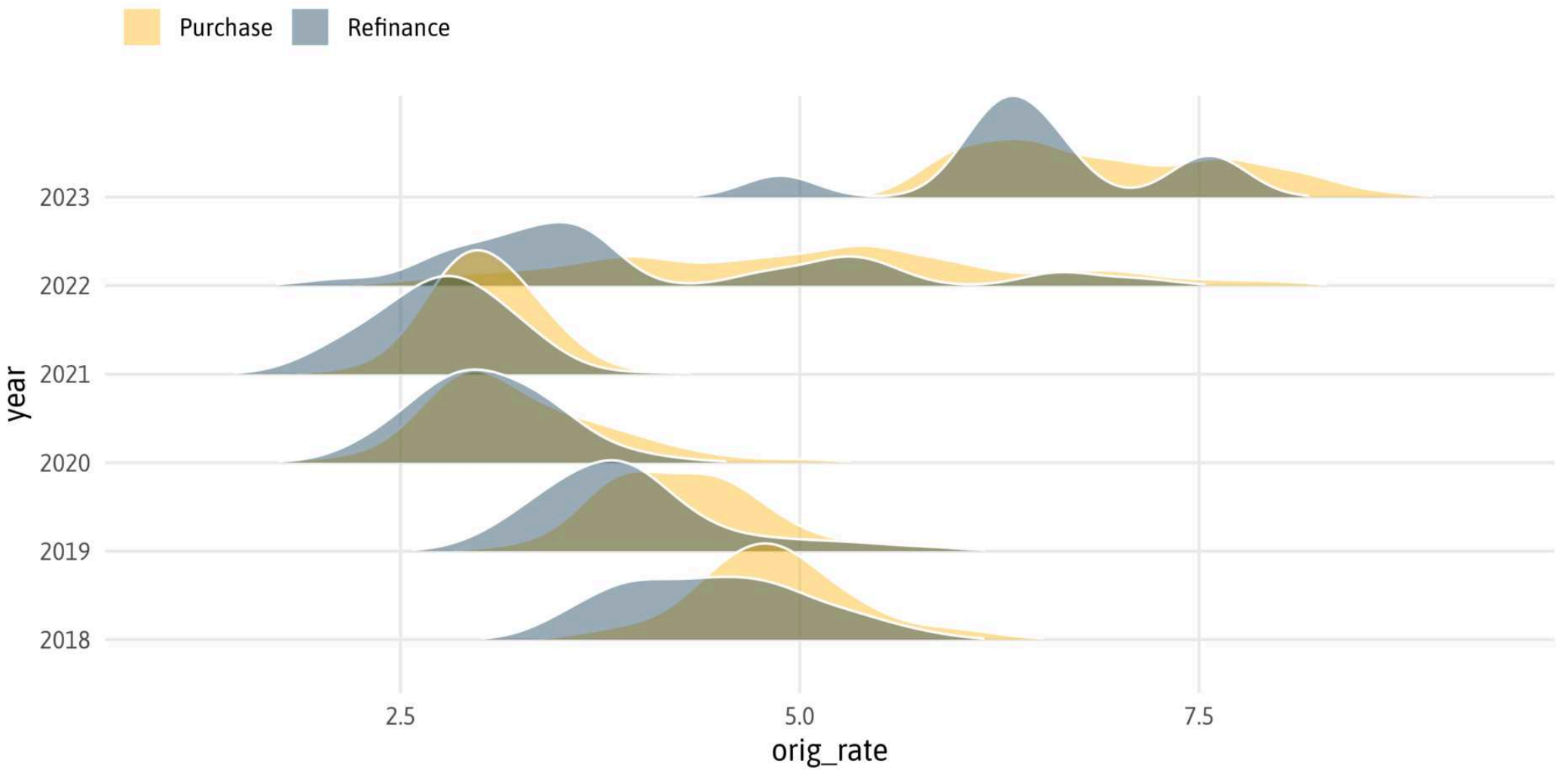
```
1 ggplot(filter(fannie_sub, first_flag == "N"), aes(x = orig_rate, y = year)) +
2   ggridges::geom_density_ridges()
```

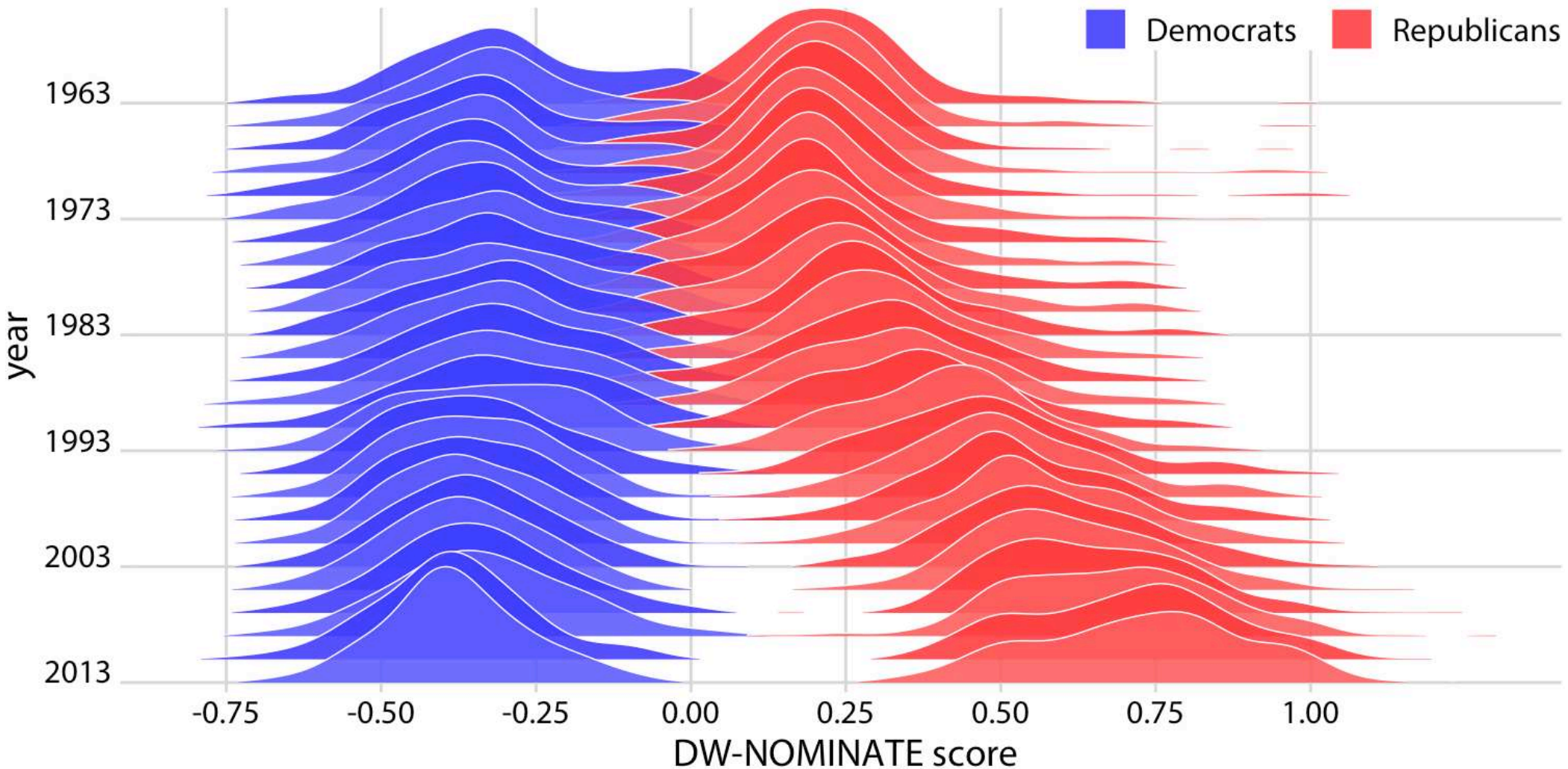


```
1 ggplot(filter(fannie_sub, first_flag == "N"), aes(x = orig_rate, y = year, fill = purp
2   ggridges::geom_density_ridges(alpha = .4)
```



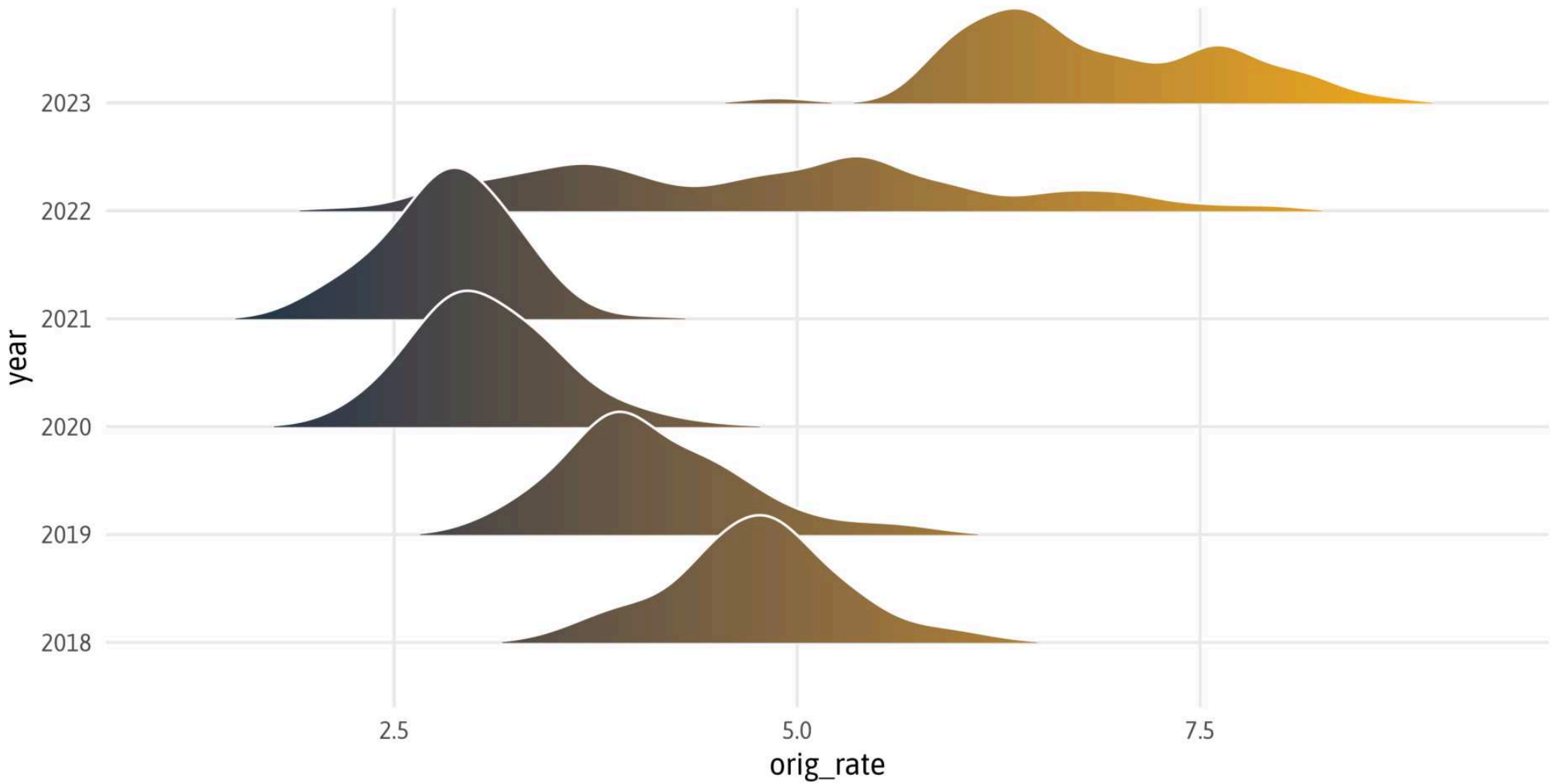
```r
ggplot(filter(fannie_sub, first_flag == "N"), aes(x = orig_rate, y = year, fill = purp
  ggridges::geom_density_ridges(alpha = .4, color = "white", scale = 1.4, rel_min_heig
  scale_fill_manual(values = c("#FFB306", "#05314D"), name = NULL)
```





Source: "Fundamentals of Data Visualization" by Claus O. Wilke; original figure by Ian McDonald

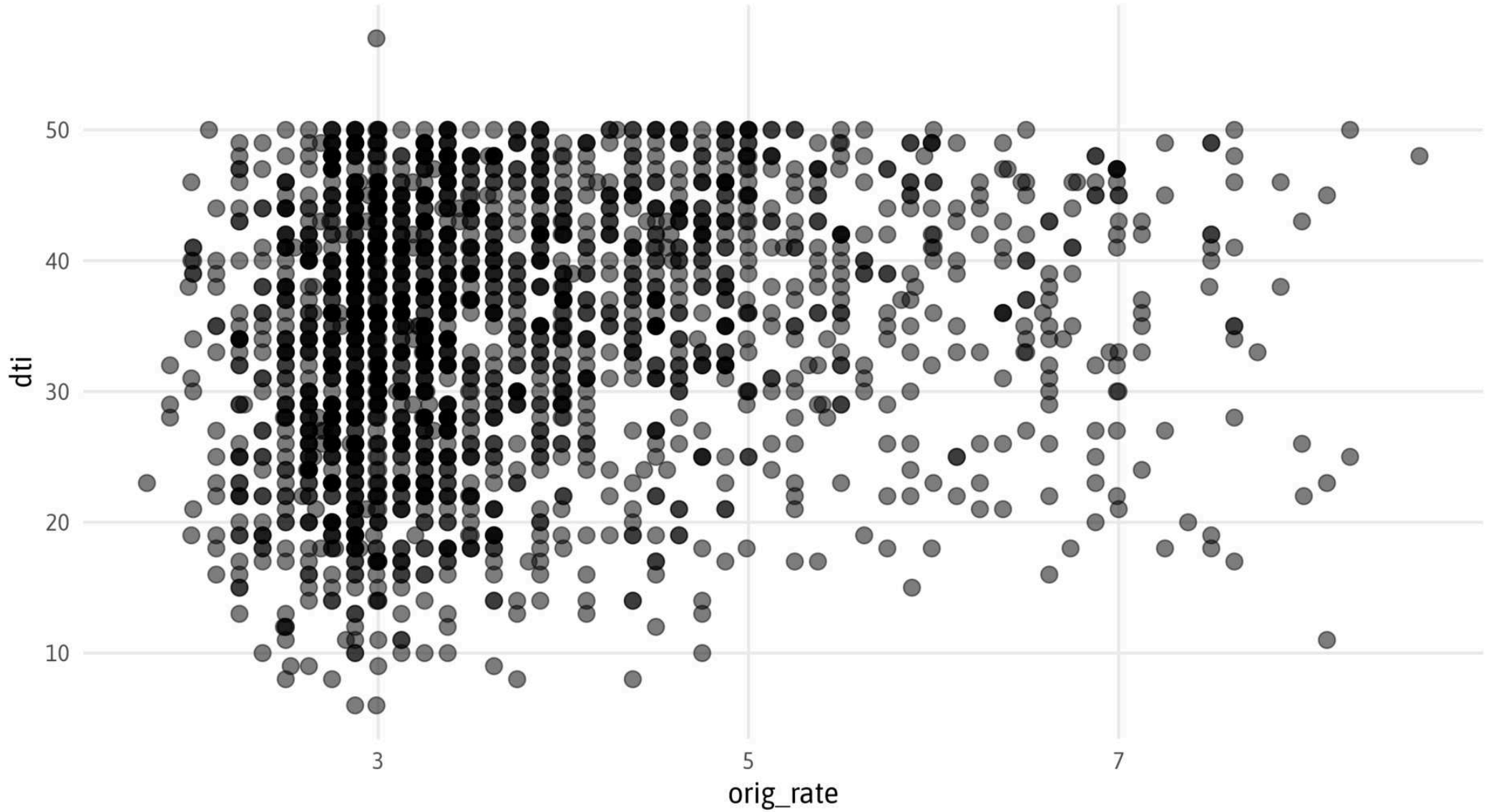
```r
ggplot(filter(fannie_sub, first_flag == "N"), aes(x = orig_rate, y = year, fill = stat
  ggridges::geom_density_ridges_gradient(color = "white", scale = 1.4, rel_min_height
  scale_fill_gradient(low = "#05314D", high = "#FFB306", guide = "none")
```



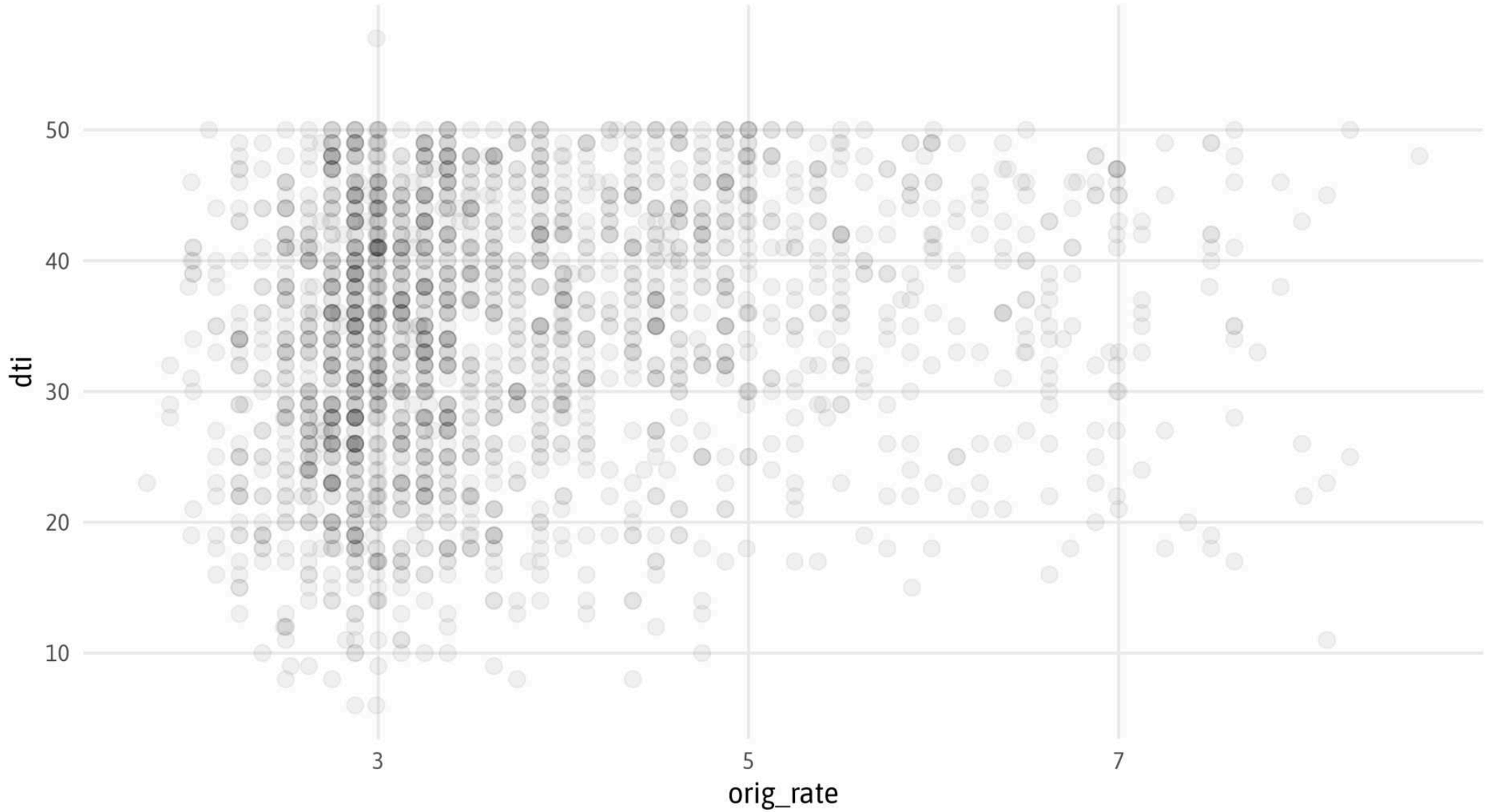
Visualizing x-y Relationships

Cédric Scherer Data Visualization & Information Design

```
1 ggplot(fannie_sub, aes(x = orig_rate, y = dti)) +
2   geom_point(size = 3, alpha = .5)
```



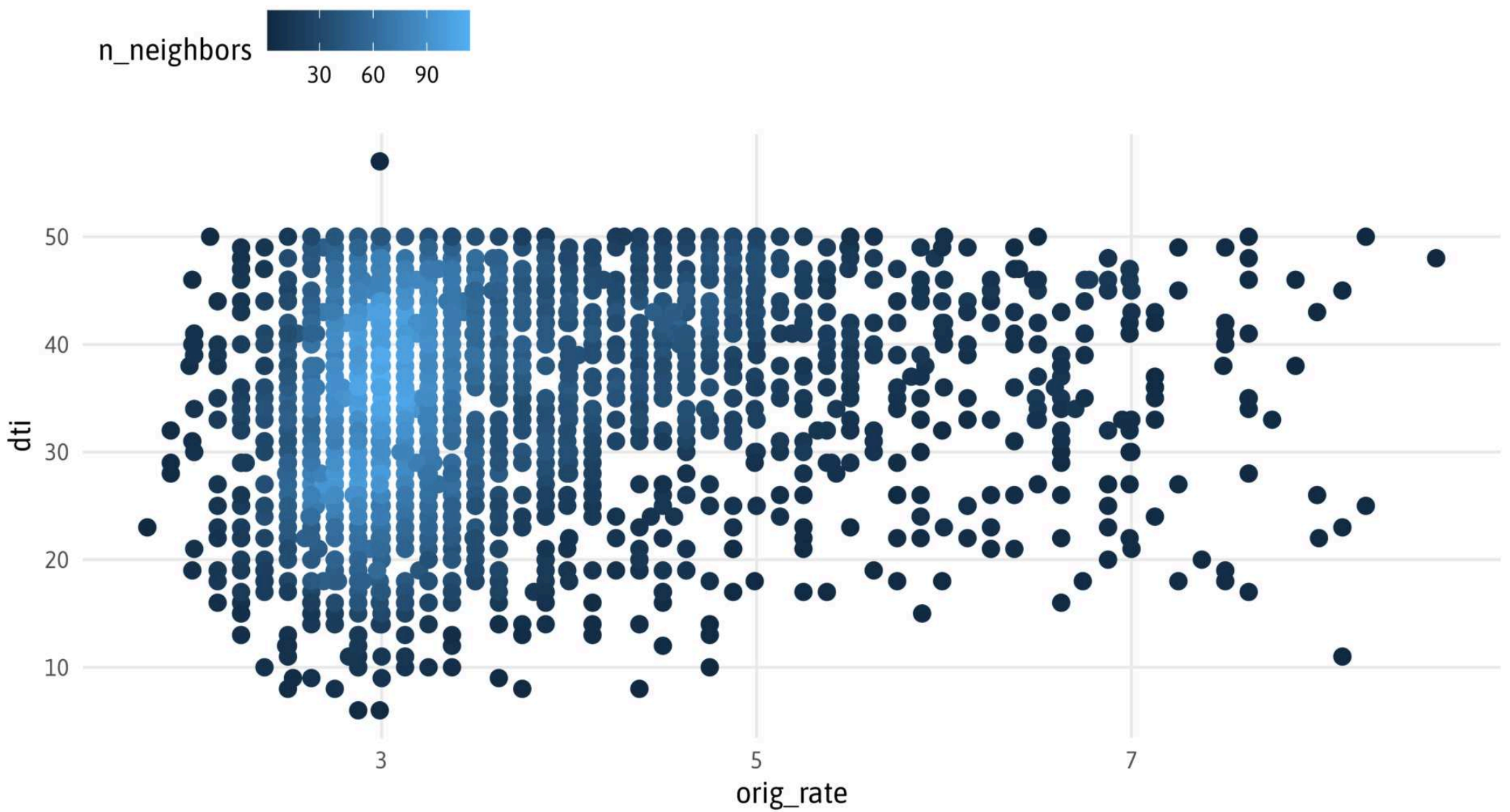
```r
ggplot(fannie_sub, aes(x = orig_rate, y = dti)) +
  geom_point(size = 3, alpha = .05)
```



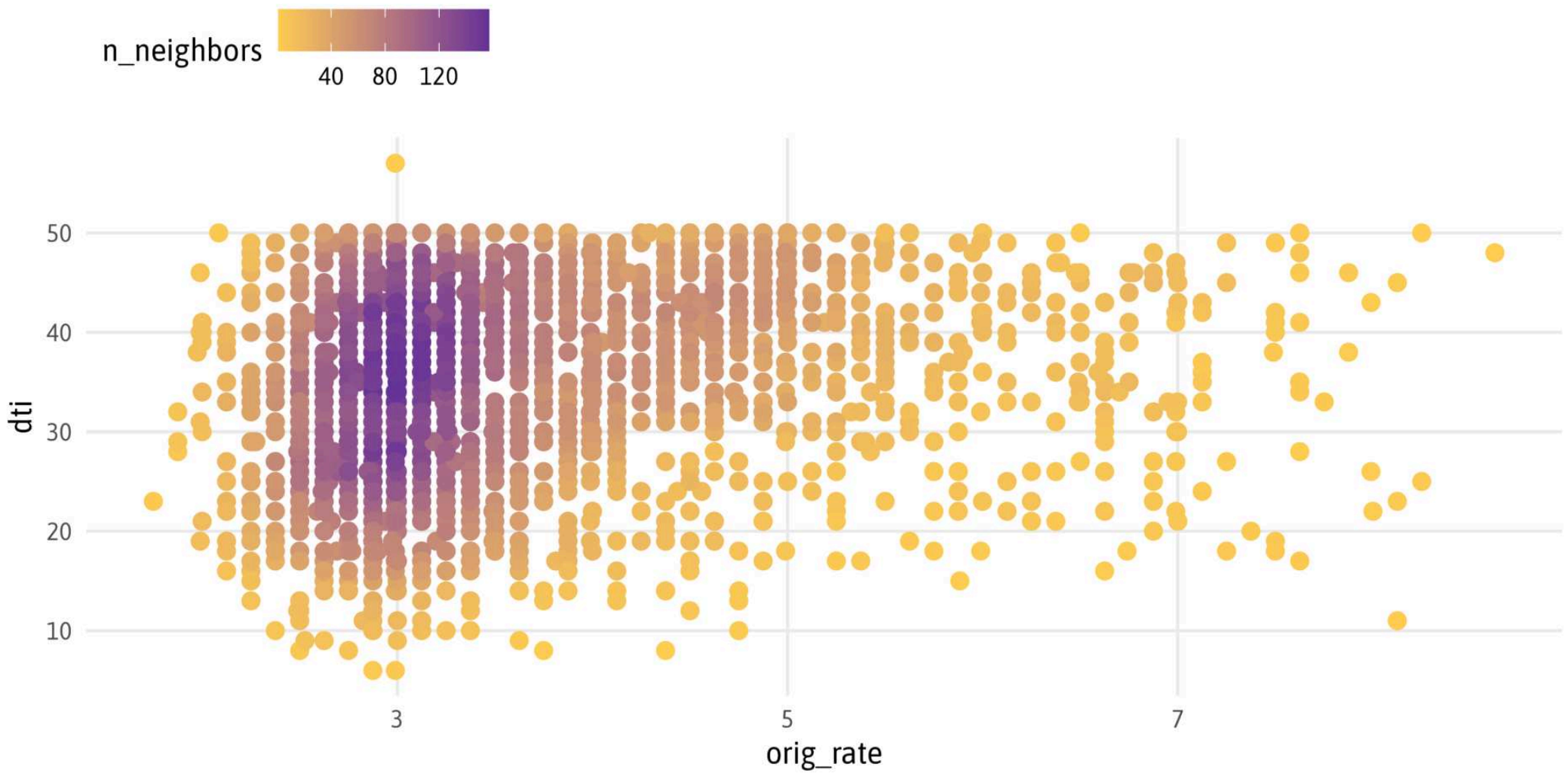
{ggpointdensity}

Cédric Scherer Data Visualization & Information Design

```
1 ggplot(fannie_sub, aes(x = orig_rate, y = dti)) +
2   ggpointdensity::geom_pointdensity(size = 3)
```



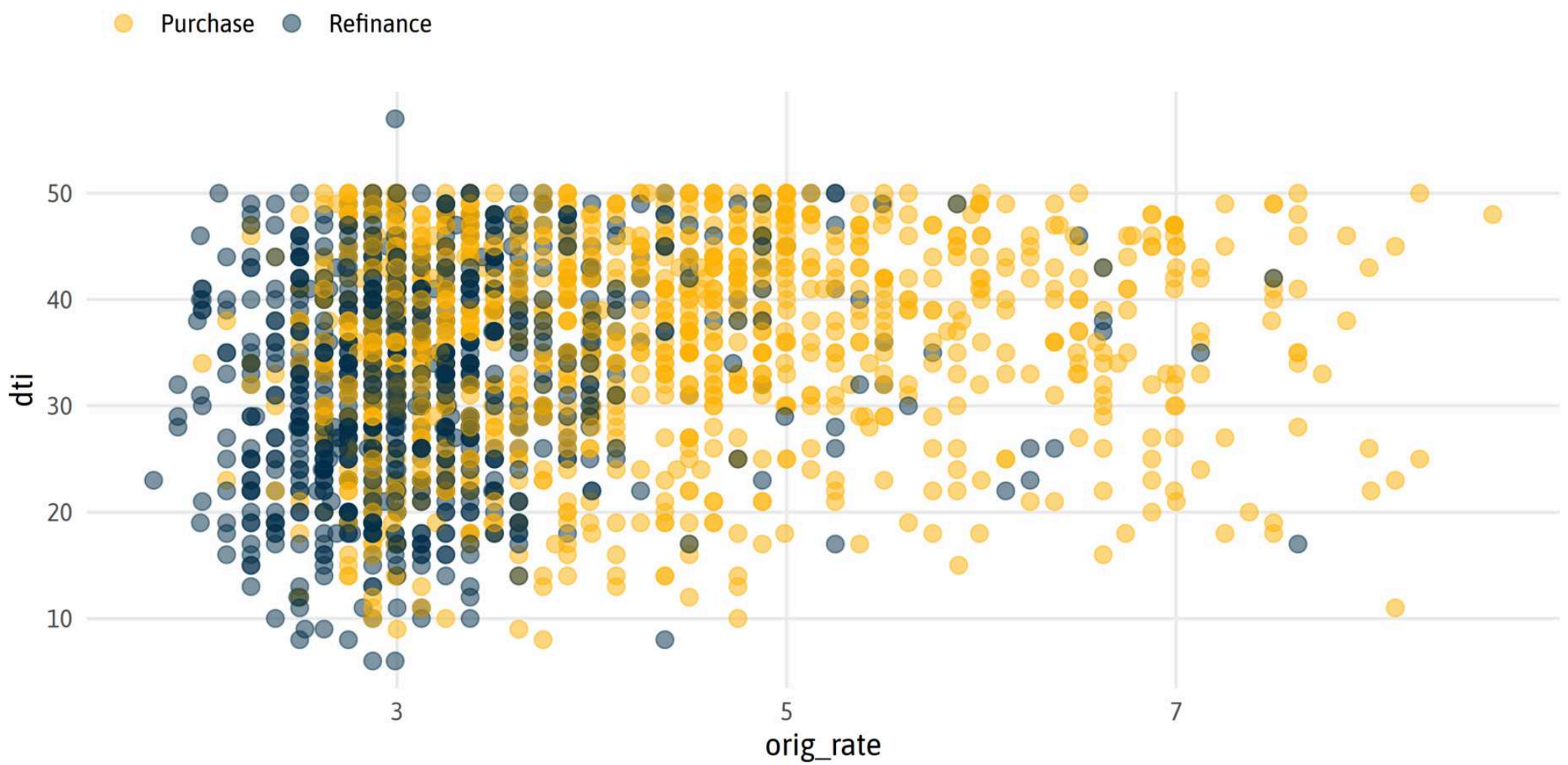
```
ggplot(fannie_sub, aes(x = orig_rate, y = dti)) +
  ggpointdensity::geom_pointdensity(size = 3, adjust = 1.5) +
  scale_color_gradient(low = "#FFCE52", high = "#663399")
```



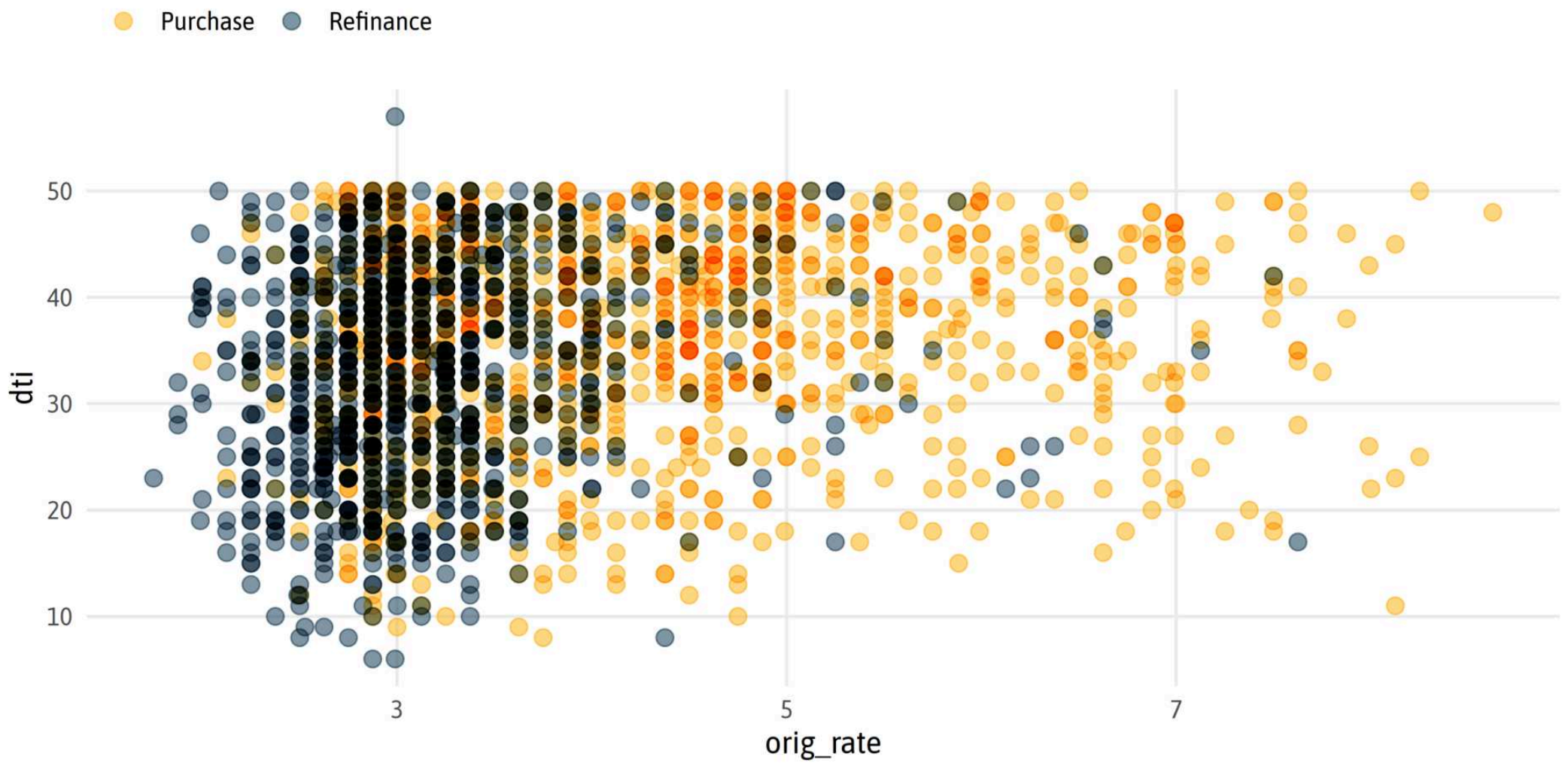
{ggblend}

Cédric Scherer Data Visualization & Information Design

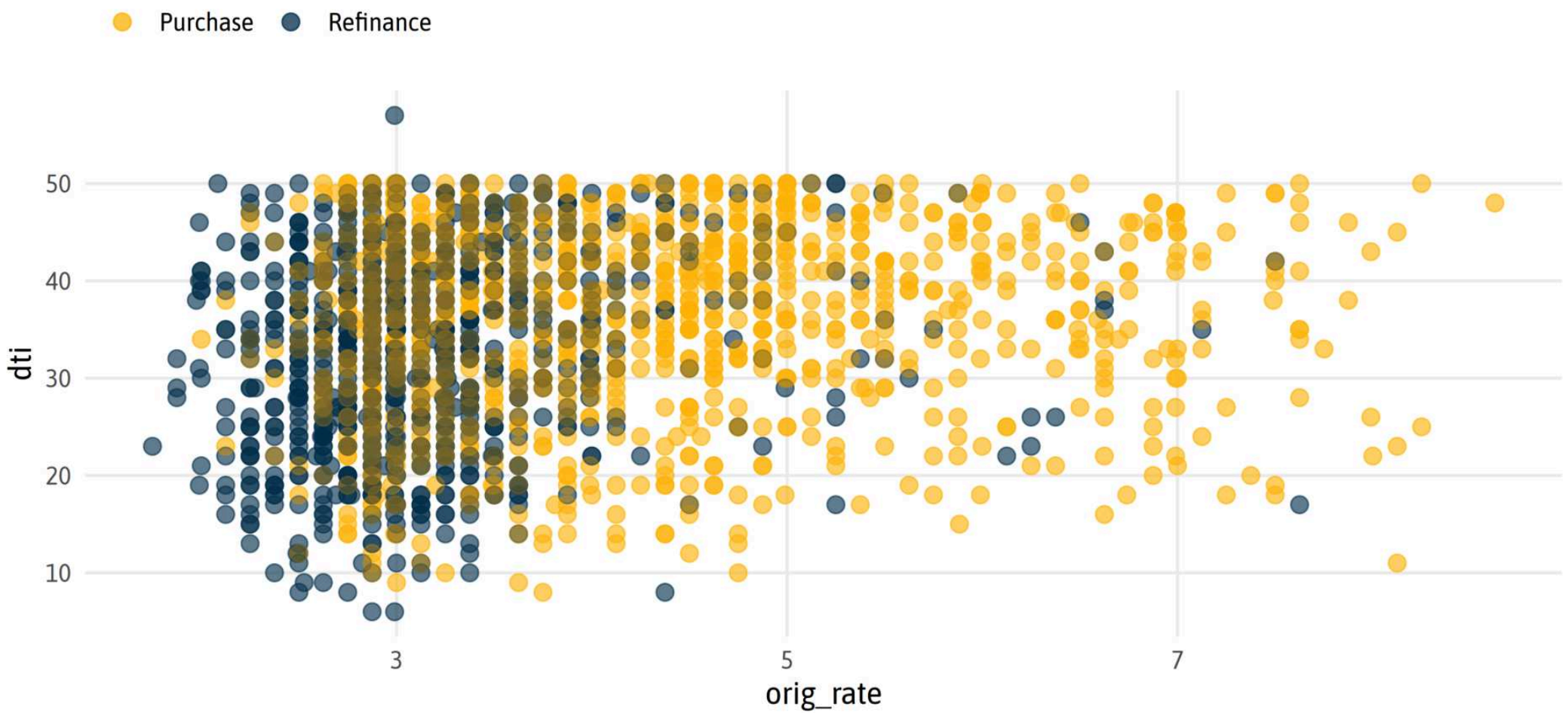
```r
ggplot(fannie_sub, aes(x = orig_rate, y = dti, color = purpose)) +
  geom_point(size = 3, alpha = .5) +
  scale_color_manual(values = c("#FFB306", "#05314D"), name = NULL)
```



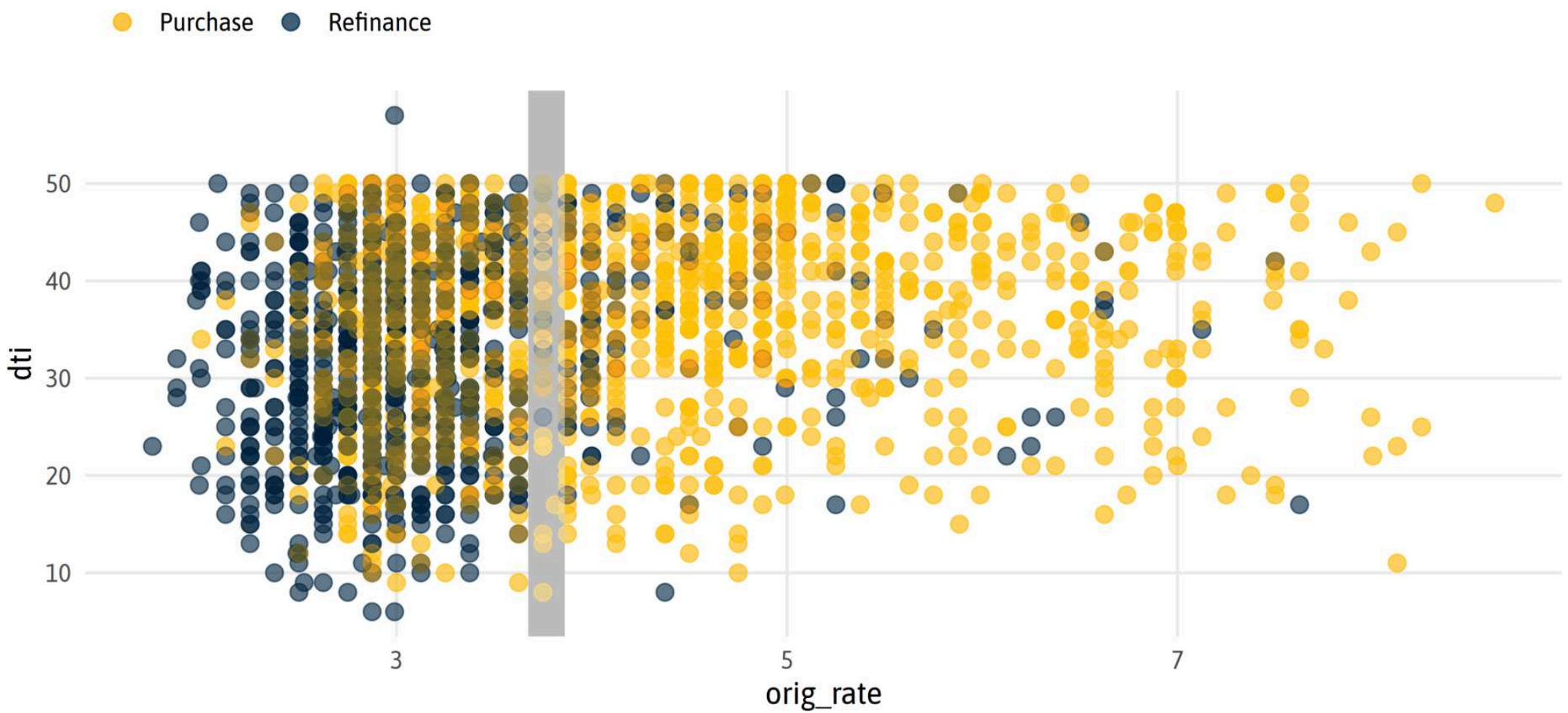
```r
ggplot(fannie_sub, aes(x = orig_rate, y = dti, color = purpose)) +
  geom_point(size = 3, alpha = .5) |> ggblend::blend("multiply") +
  scale_color_manual(values = c("#FFB306", "#05314D"), name = NULL)
```



```r
library(ggblend)
ggplot(fannie_sub, aes(x = orig_rate, y = dti, color = purpose, partition = purpose)) +
  geom_point(size = 3, alpha = .5) * (blend("lighten") + blend("multiply", alpha = 0.5
  scale_color_manual(values = c("#FFB306", "#05314D"), name = NULL)
```



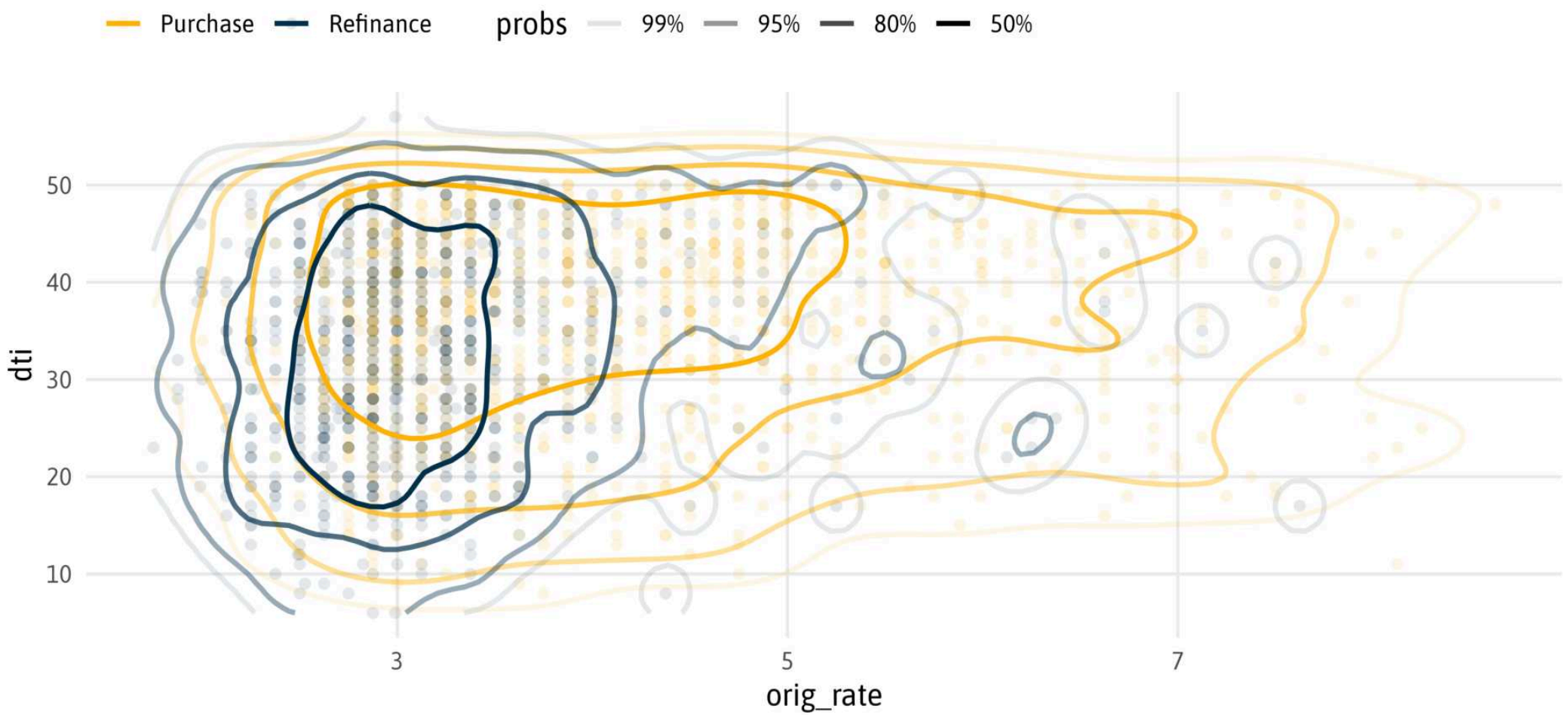
```r
ggplot(fannie_sub, aes(x = orig_rate, y = dti, color = purpose, partition = purpose)) +
  list(geom_point(size = 3, alpha = .5) * (blend("lighten") + blend("multiply", alpha =
       geom_vline(xintercept = mean(fannie_sub$orig_rate), color = "grey", linewidth =
  scale_color_manual(values = c("#FFB306", "#05314D"), name = NULL)
```



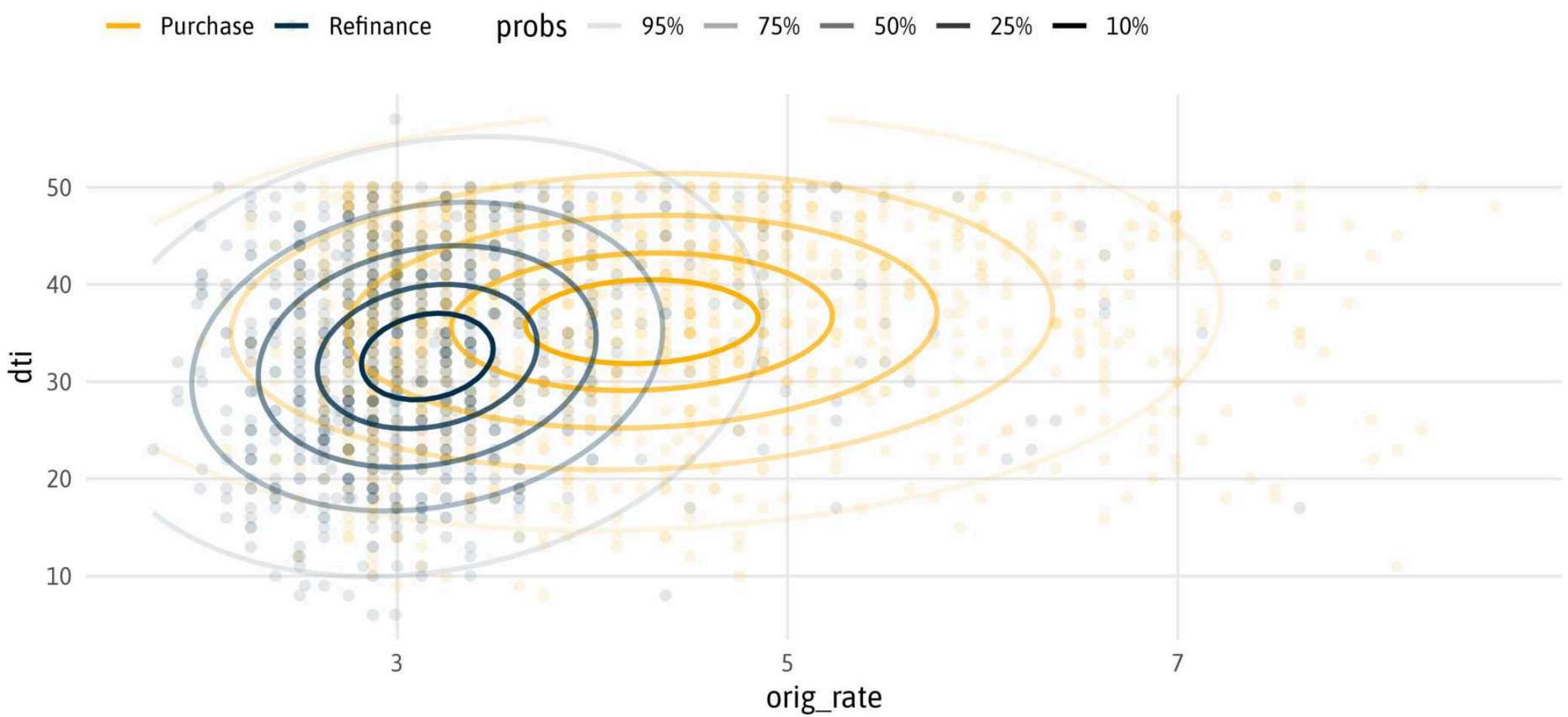
{ggdensity}

Cédric Scherer Data Visualization & Information Design

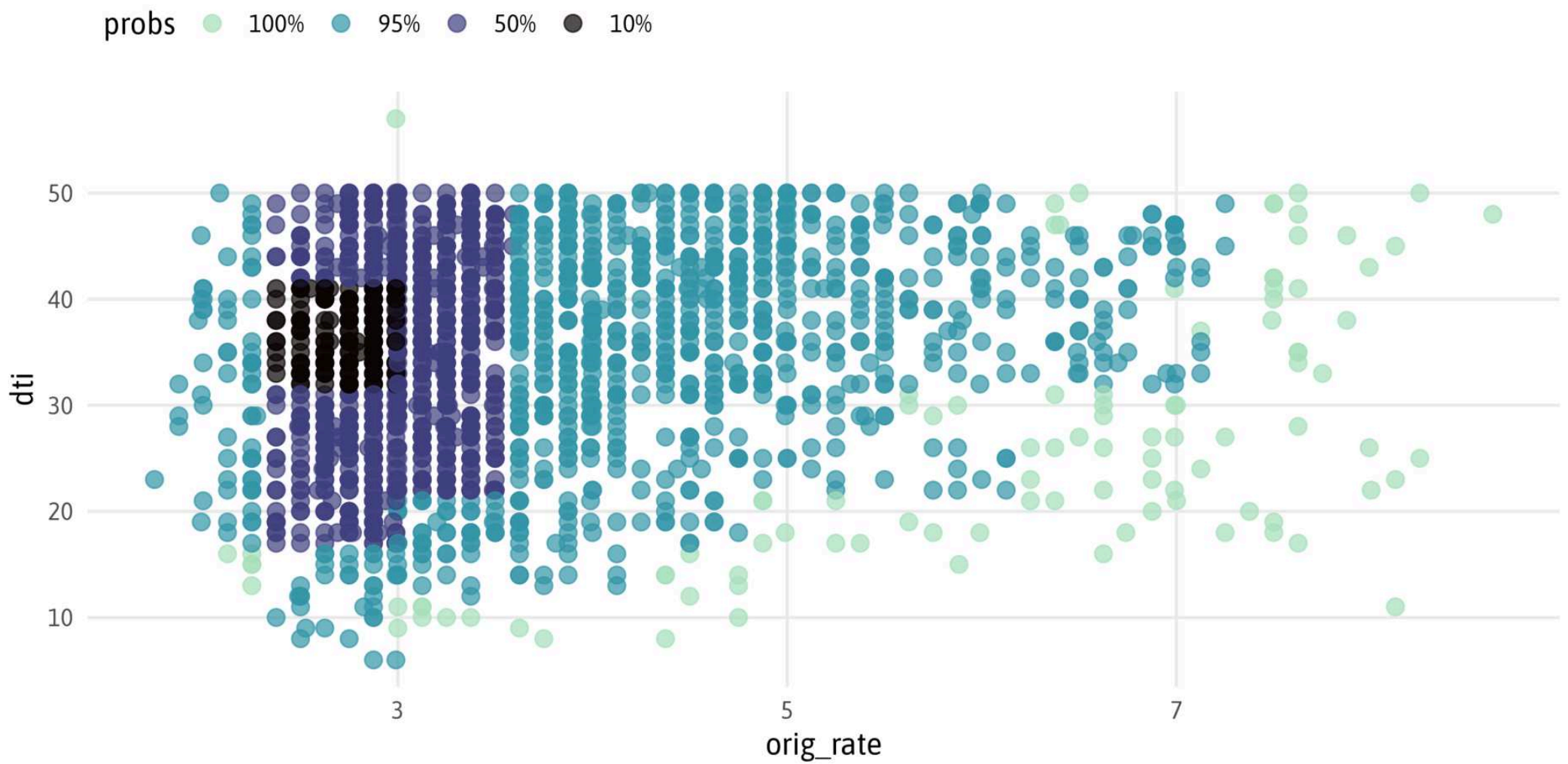
```
ggplot(fannie_sub, aes(x = orig_rate, y = dti, color = purpose)) +
  geom_point(alpha = .1, shape = 16, size = 2) +
  ggdensity::geom_hdr_lines() +
  scale_color_manual(values = c("#FFB306", "#05314D"), name = NULL)
```



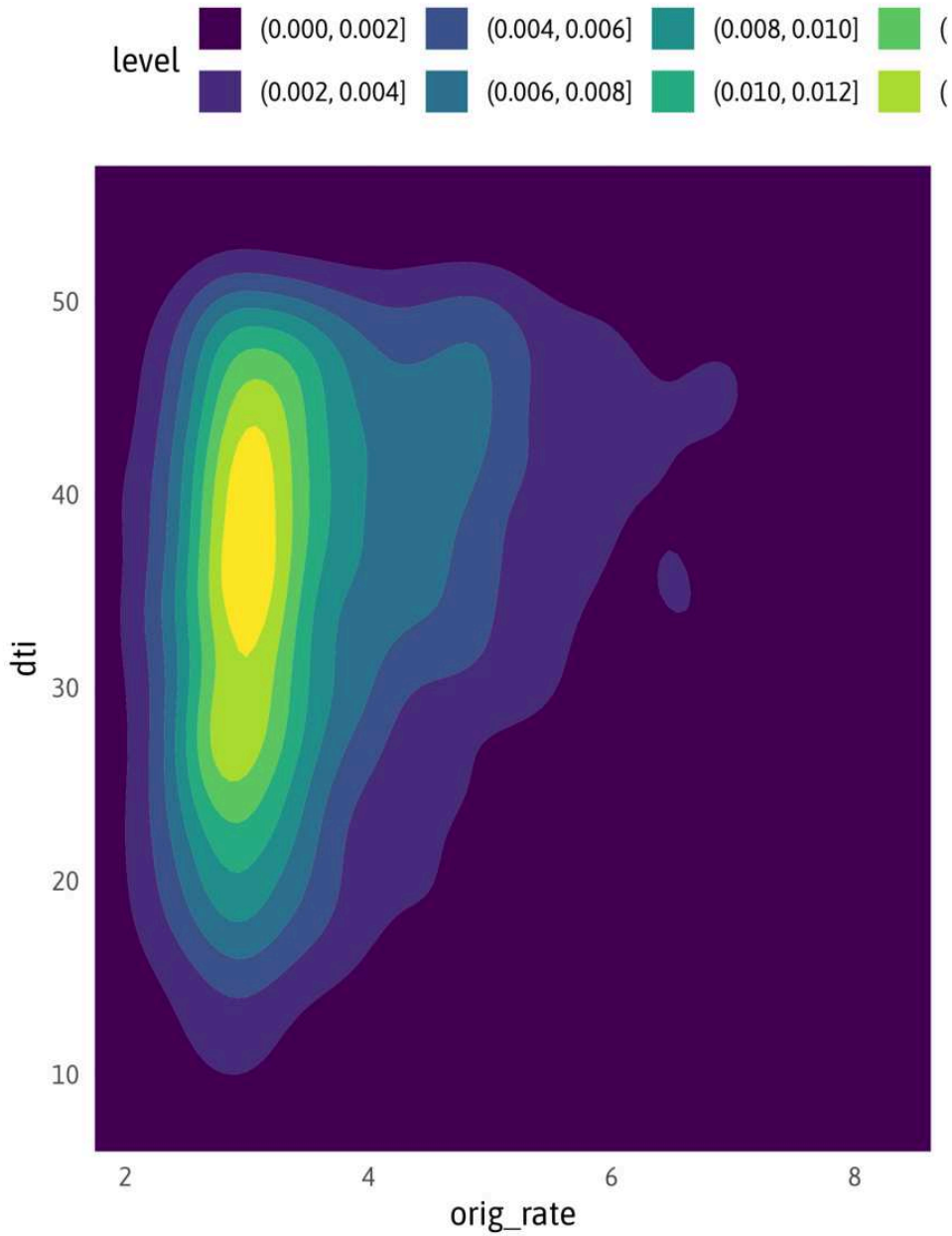
```
ggplot(fannie_sub, aes(x = orig_rate, y = dti, color = purpose)) +
  geom_point(alpha = .1, shape = 16, size = 2) +
  ggdensity::geom_hdr_lines(method = "mvnorm", probs = c(.95, .75, .5, .25, .1)) +
  scale_color_manual(values = c("#FFB306", "#05314D"), name = NULL)
```



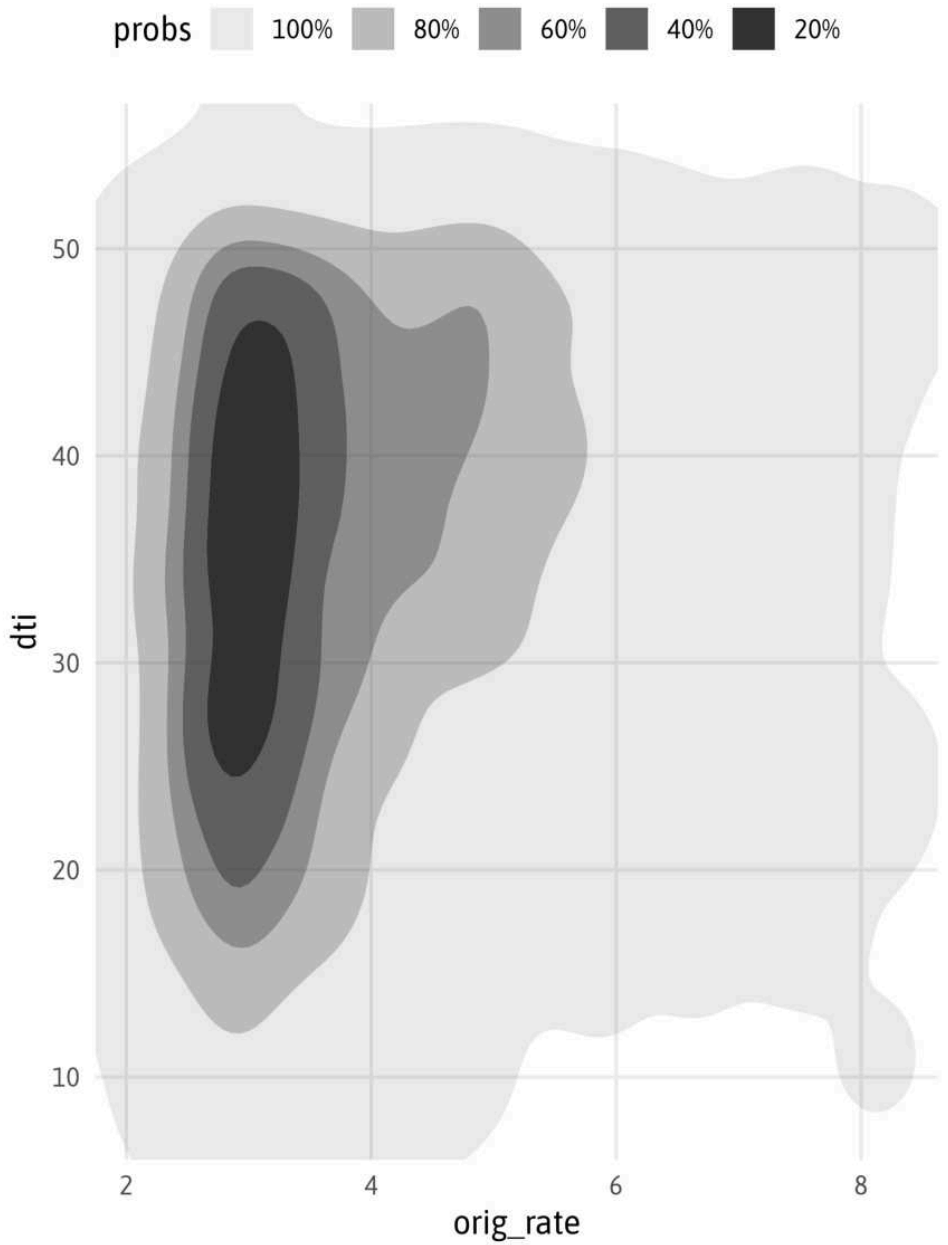
```r
ggplot(fannie_sub, aes(x = orig_rate, y = dti)) +
  ggdensity::geom_hdr_points(method = "histogram", probs = c(.95, .5, .1), size = 3, a
  scale_color_viridis_d(option = "mako", direction = -1, end = .9)
```



geom_density_2d_filled()



ggdensity::geom_hdr()

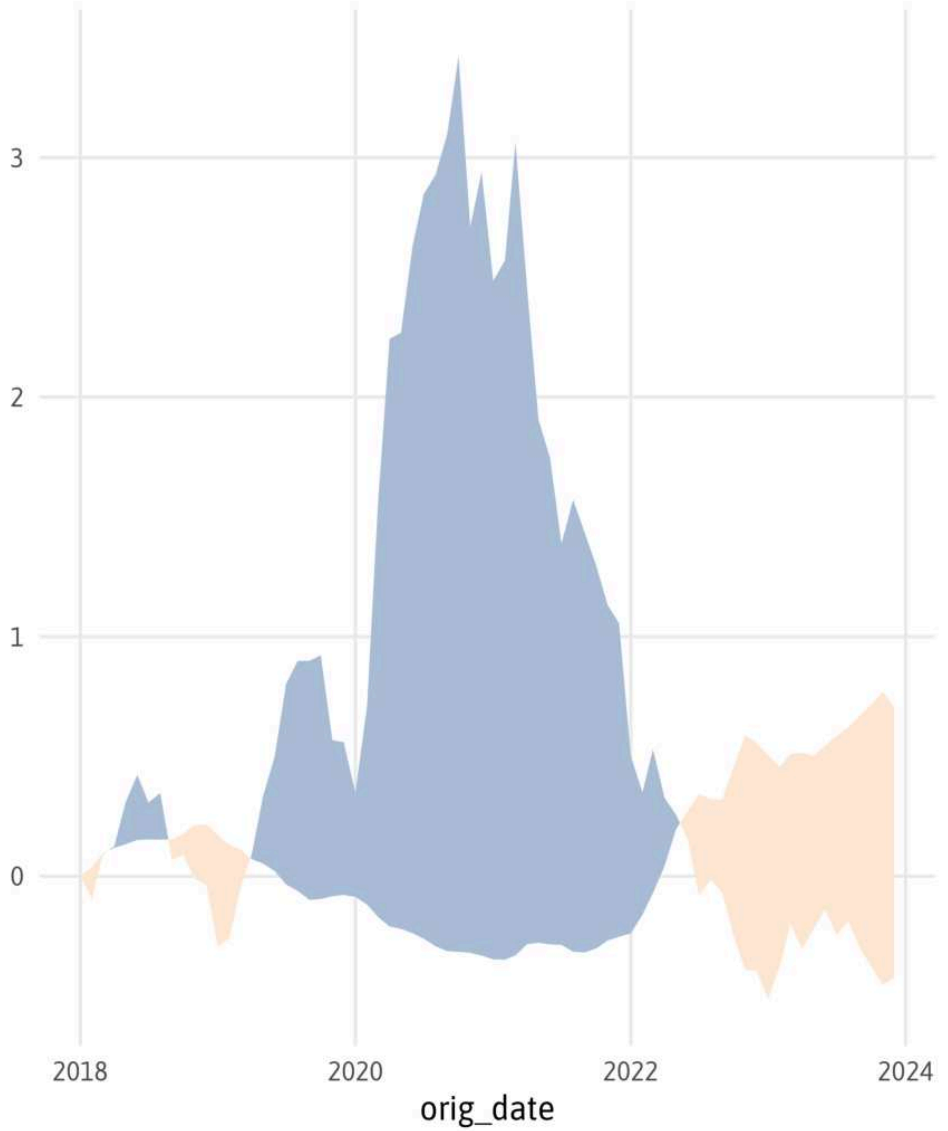


{ggbraid}

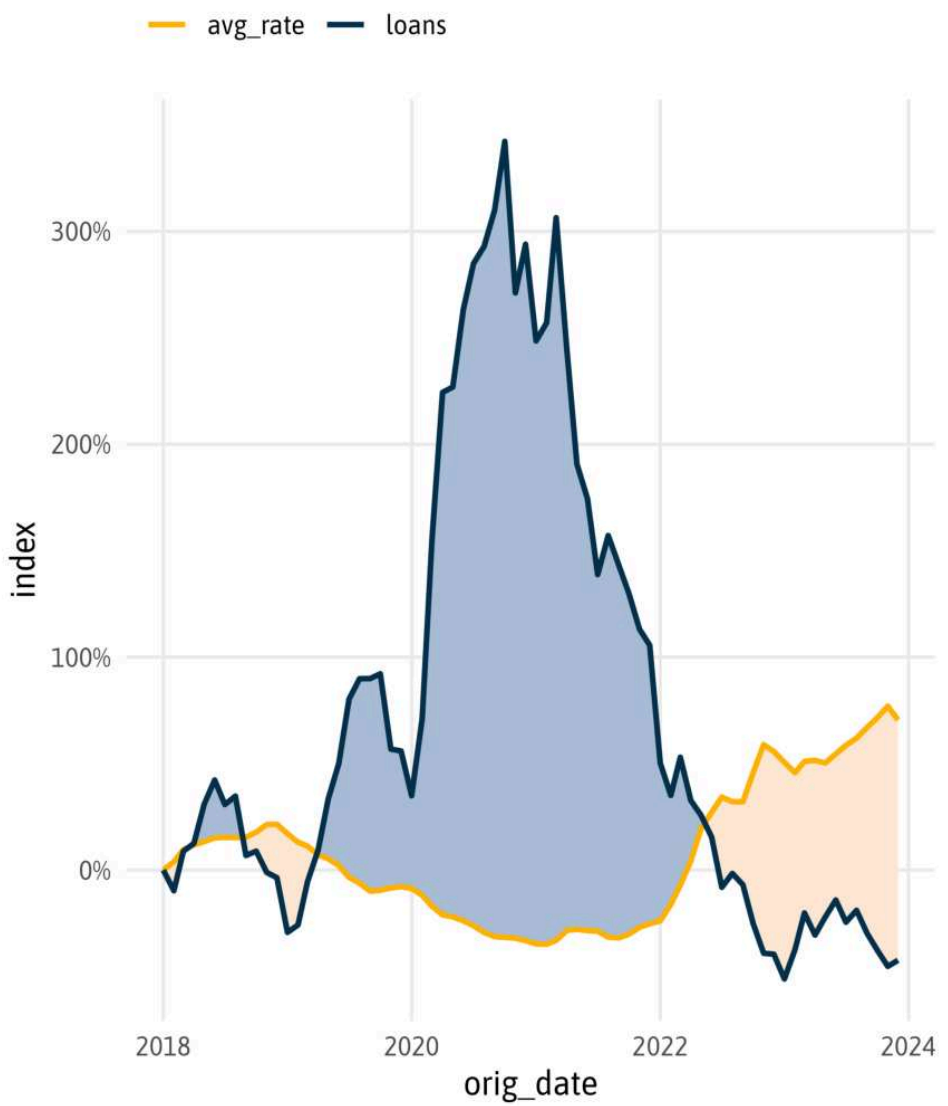
Cédric Scherer Data Visualization & Information Design

```r
fannie_index <-
  fannie_raw |>
  summarize(
    loans = n(),
    avg_rate = mean(orig_rate),
    .by = orig_date
  ) |>
  mutate(
    loans = loans / first(loans) - 1,
    avg_rate = avg_rate / first(avg_rate) - 1
  )

fannie_index_long <- tidyr::pivot_longer(
  fannie_index, cols = -c(orig_date),
  names_to = "variable", values_to = "index"
)
```

```r
ggplot(fannie_index,
       aes(x = orig_date)) +
  ggbraid::geom_braid(
    aes(ymin = loans,
        ymax = avg_rate,
        fill = loans > avg_rate)
  ) +
  scale_fill_manual(
    values = c("#FFE8D3", "#A9BDD6'
    guide = "none"
  )
```



```r
ggplot(fannie_index,
       aes(x = orig_date)) +
  ggbraid::geom_braid(
    aes(ymin = loans,
        ymax = avg_rate,
        fill = loans > avg_rate)
  ) +
  geom_line(
    data = fannie_index_long,
    aes(y = index, color = variable
    linewidth = 1
  ) +
  scale_y_continuous(
    labels = scales::percent
  ) +
  scale_color_manual(
    values = c("#FFB306", "#05314D'
    name = NULL
  ) +
  scale_fill_manual(
    values = c("#FFE8D3", "#A9BDD6'
    guide = "none"
  )
```



Composing ggplots

Cédric Scherer Data Visualization & Information Design

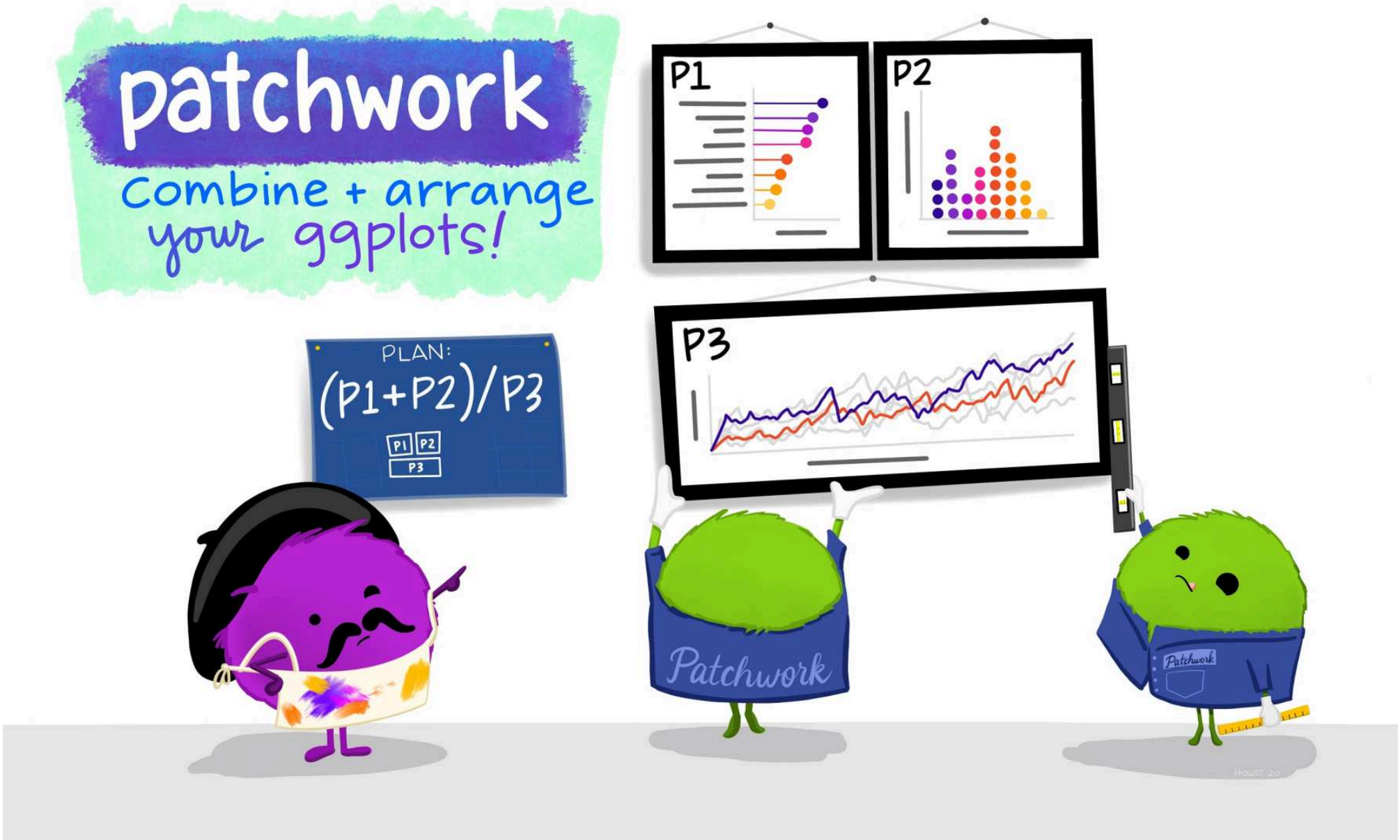
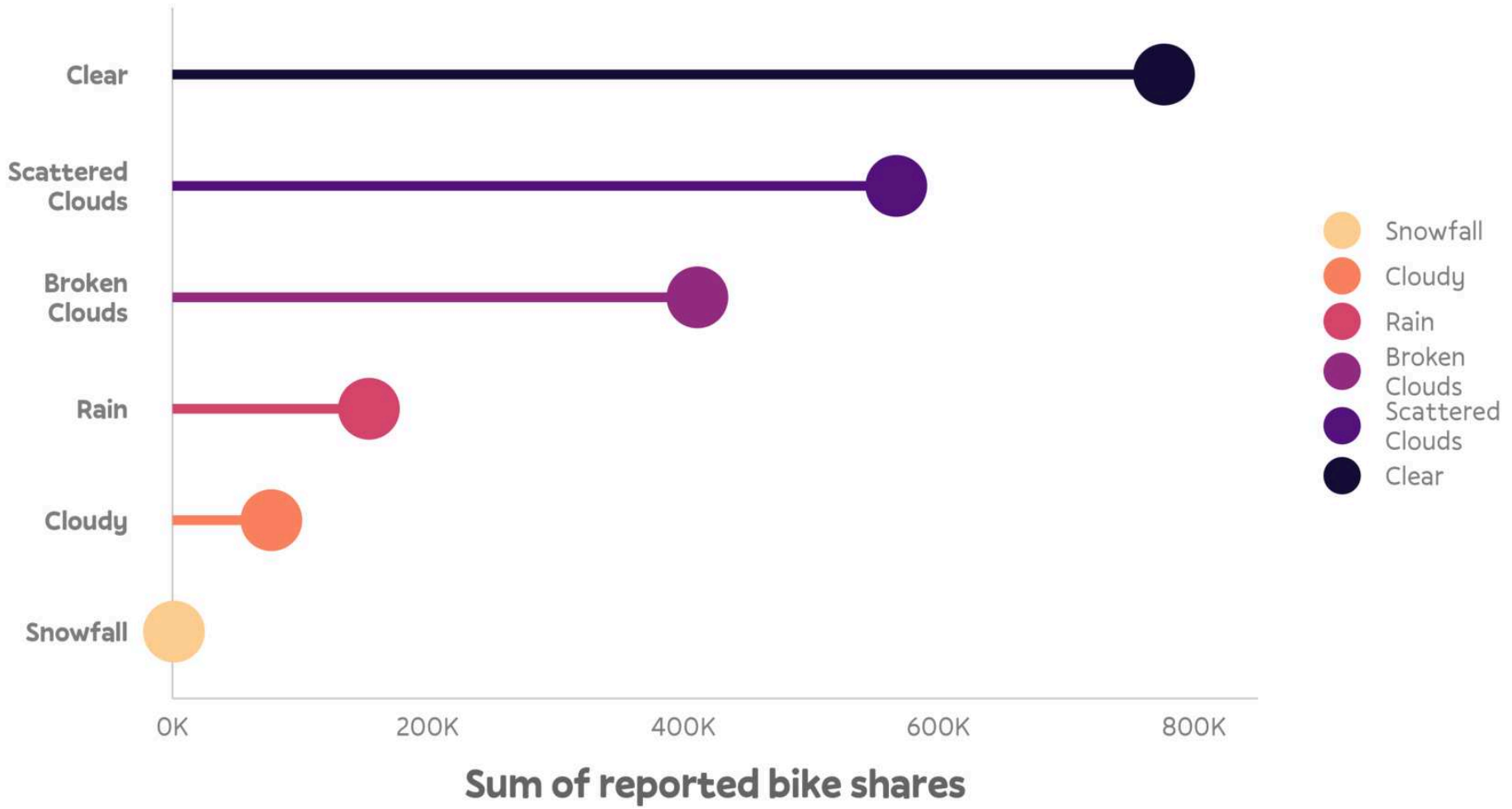


Illustration by Allison Horst

Cédric Scherer Data Visualization & Information Design

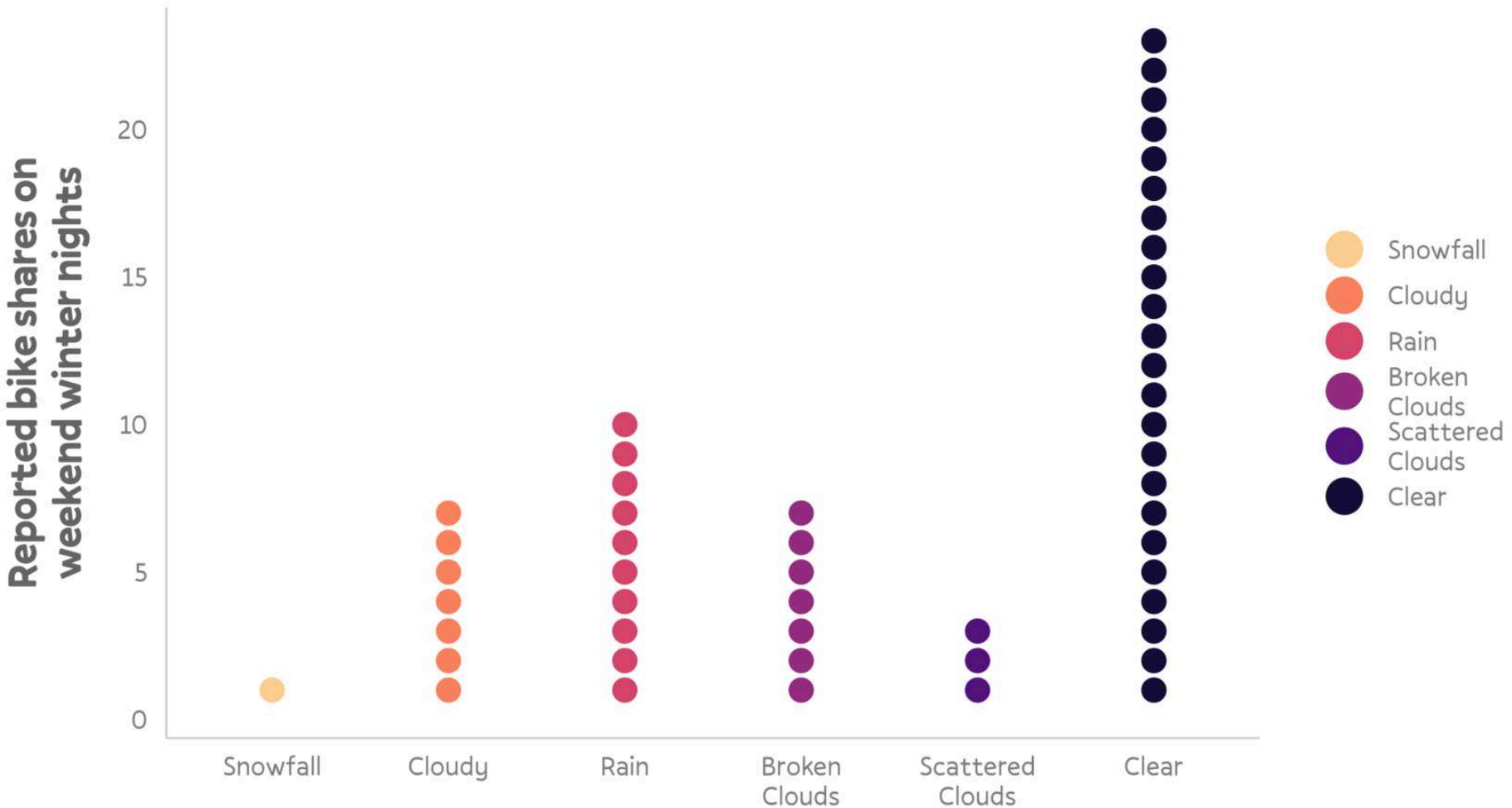
Graphic | Code

P1



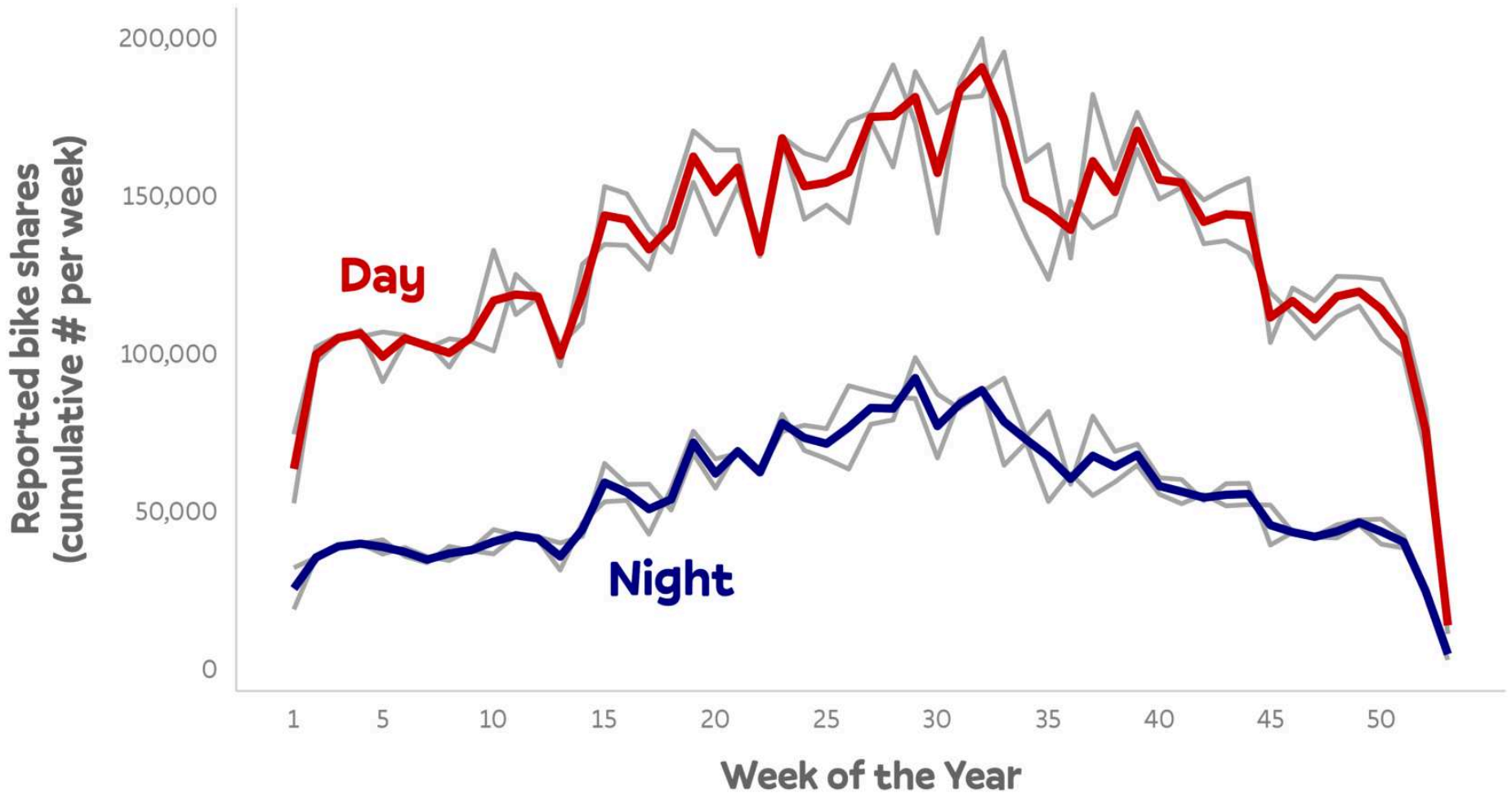
Graphic | Code

P2



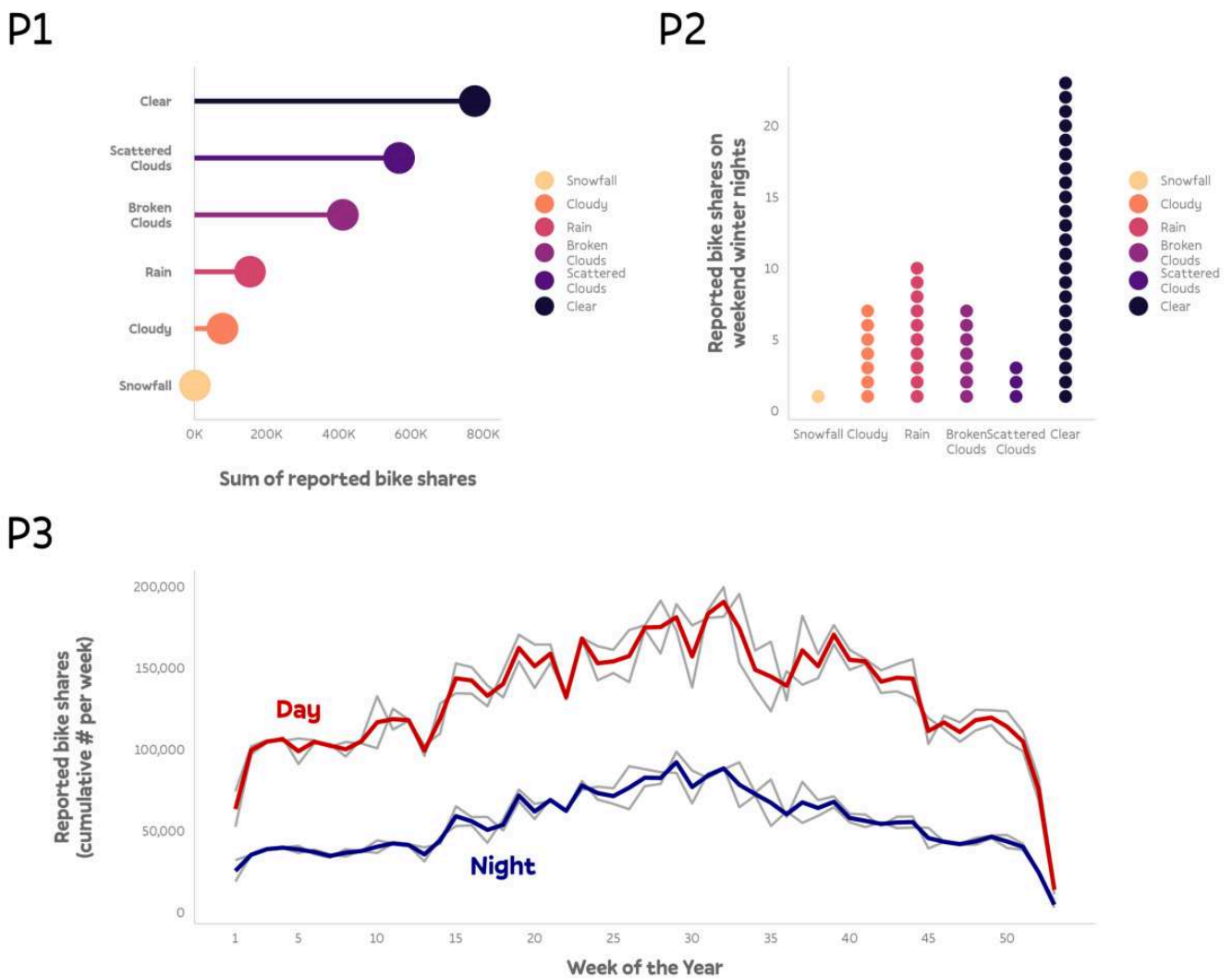
Graphic | Code

P3



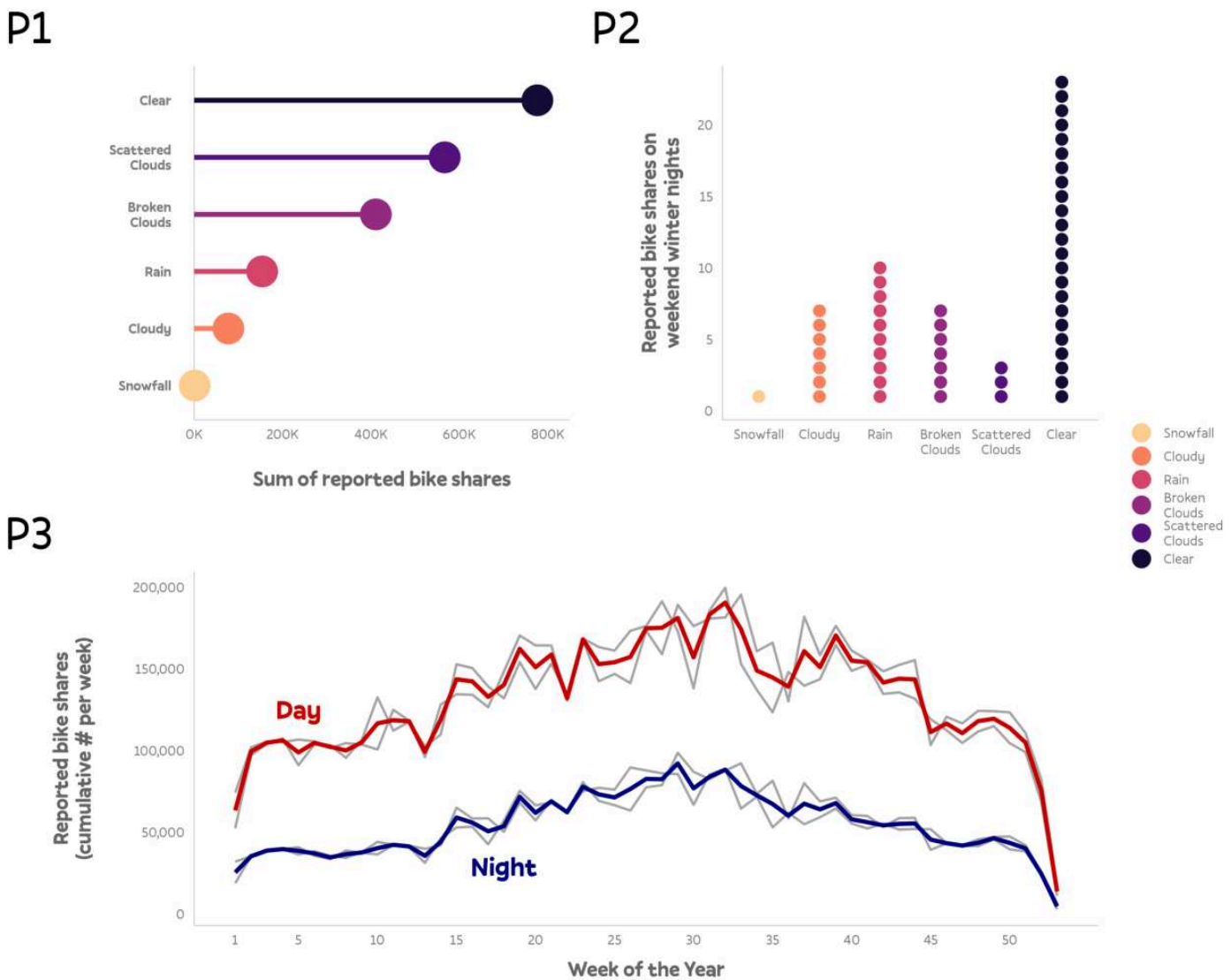
{patchwork}

```
1 library(patchwork)
2 (p1 + p2) / p3
```



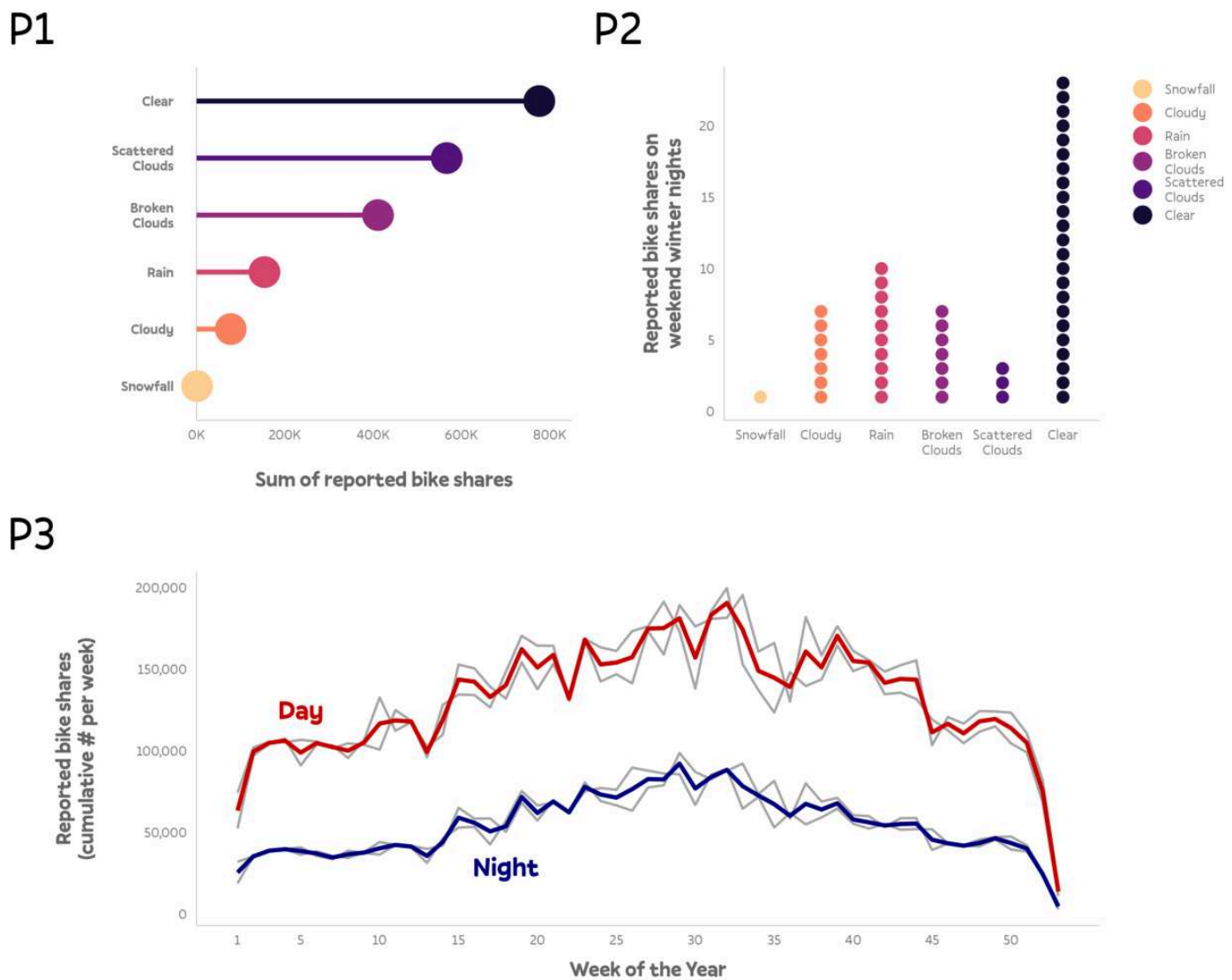
"Collect Guides"

```
1  (p1 + p2) / p3 + plot_layout(guides = "collect")
```



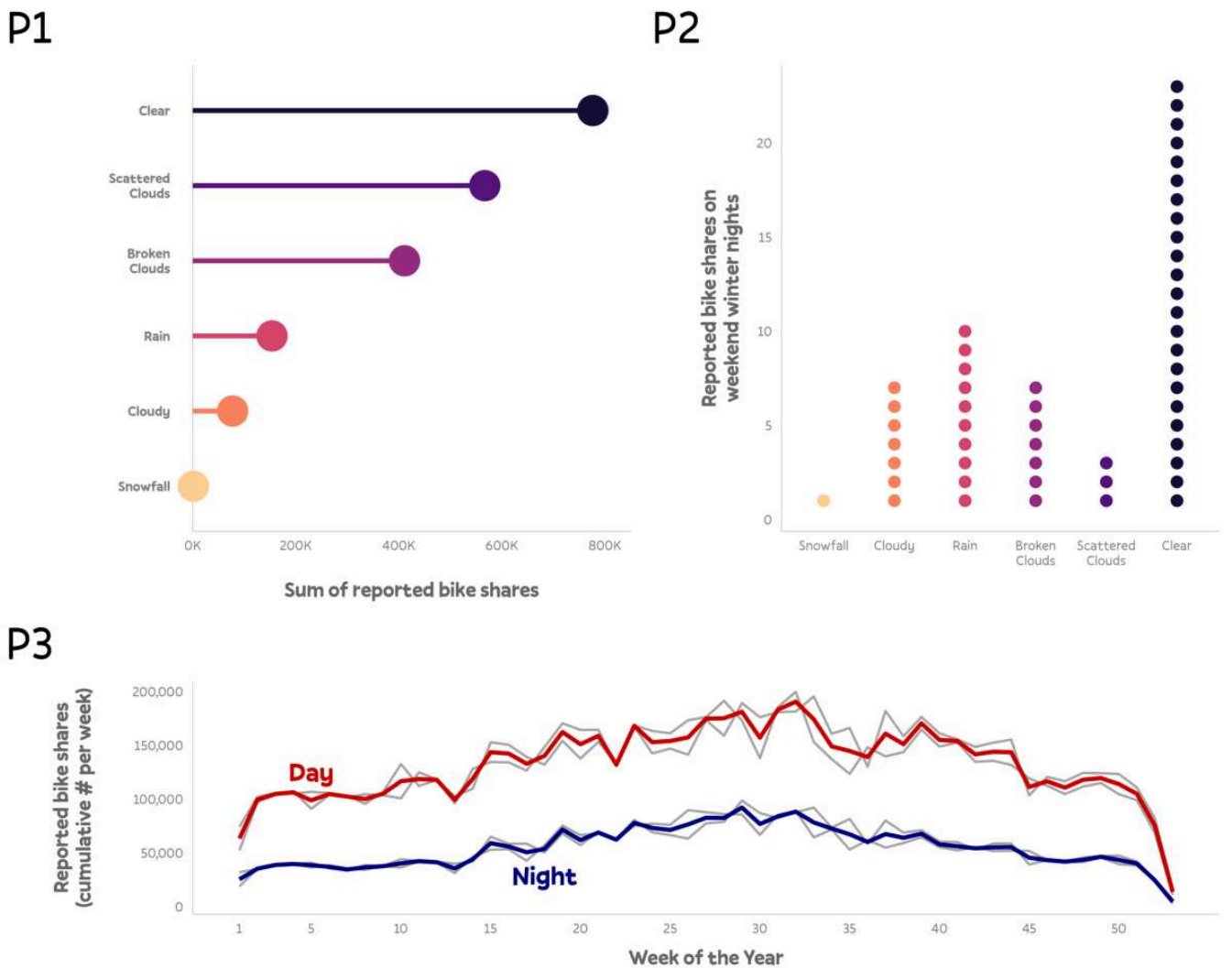
Apply Theming

```
1  ((p1 + p2) / p3 & theme(legend.justification = "top")) + plot_layout(guides = "collect
```



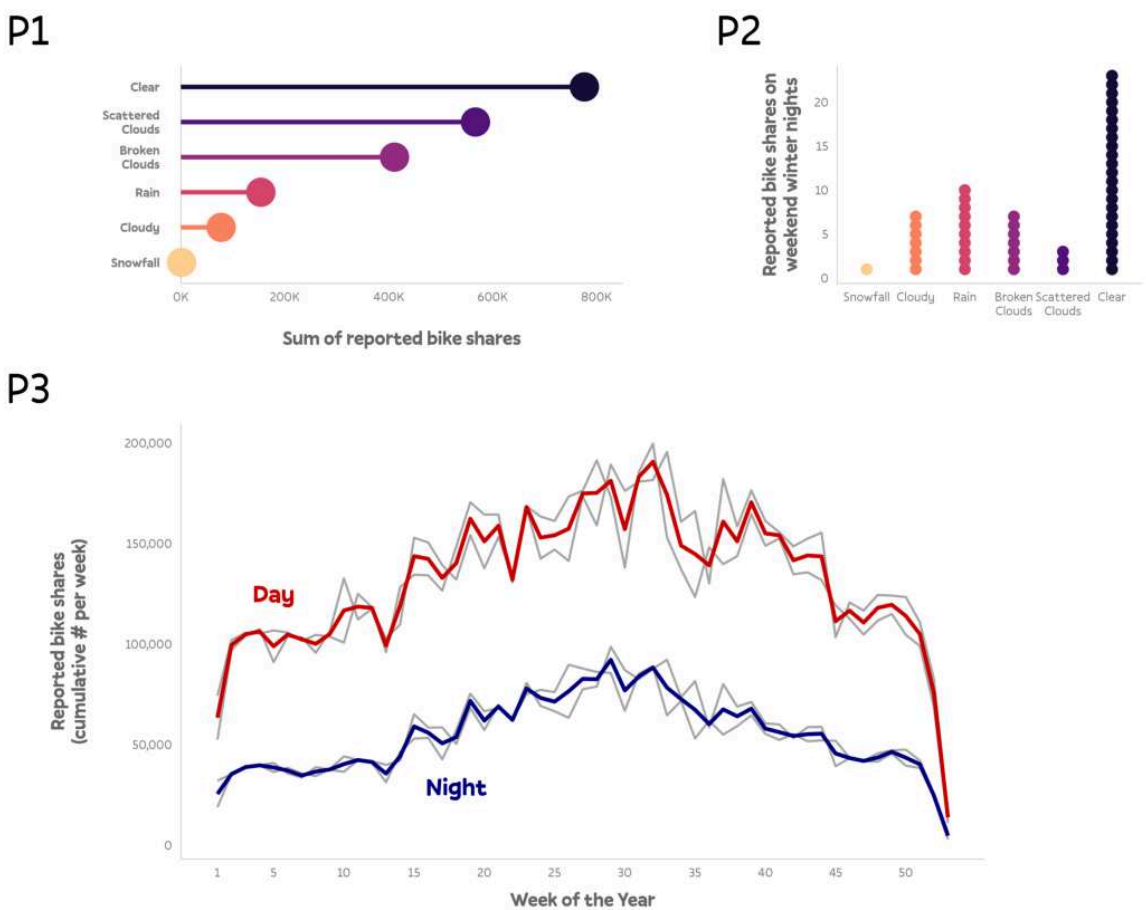
Adjust Widths and Heights

```
((p1 + p2) / p3 & theme(legend.position = "none")) +
  plot_layout(heights = c(.2, .1), widths = c(2, 1))
```



Use A Custom Layout

```
1  picasso <- "
2  AAAAAA#BBBB
3  CCCCCCCCC##
4  CCCCCCCCC##"
5  (p1 + p2 + p3 & theme(legend.position = "none")) + plot_layout(design = picasso)
```

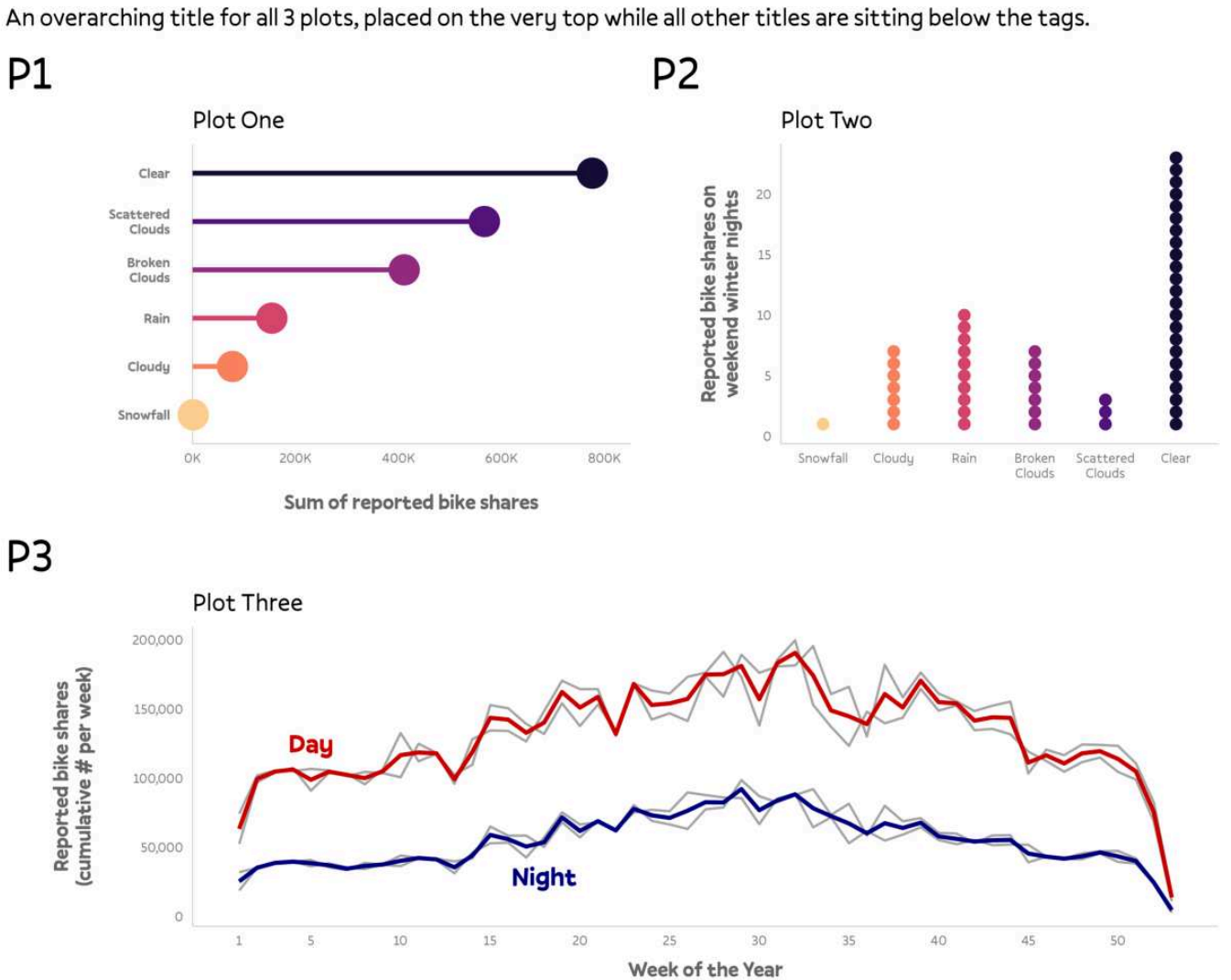


Add Labels

```
pl1 <- p1 + labs(tag = NULL, title = "Plot One") + theme(legend.position = "none")
pl2 <- p2 + labs(tag = NULL, title = "Plot Two") + theme(legend.position = "none")
pl3 <- p3 + labs(tag = NULL, title = "Plot Three") + theme(legend.position = "none")
```

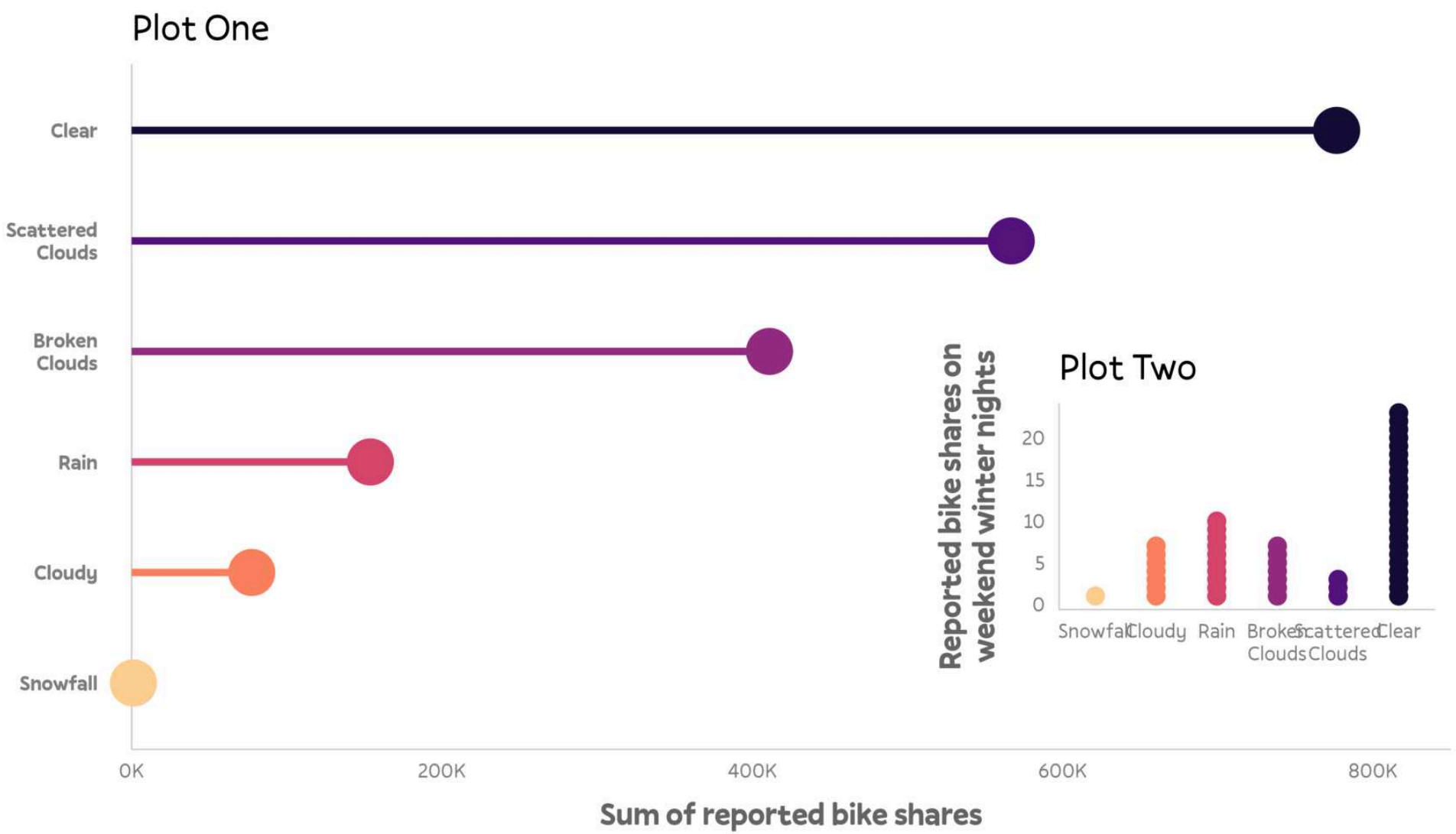
Add Labels

```
1  (pl1 + pl2) / pl3 +
2    plot_annotation(tag_levels = "1", tag_prefix = "P", title = "An overarching title fo
```



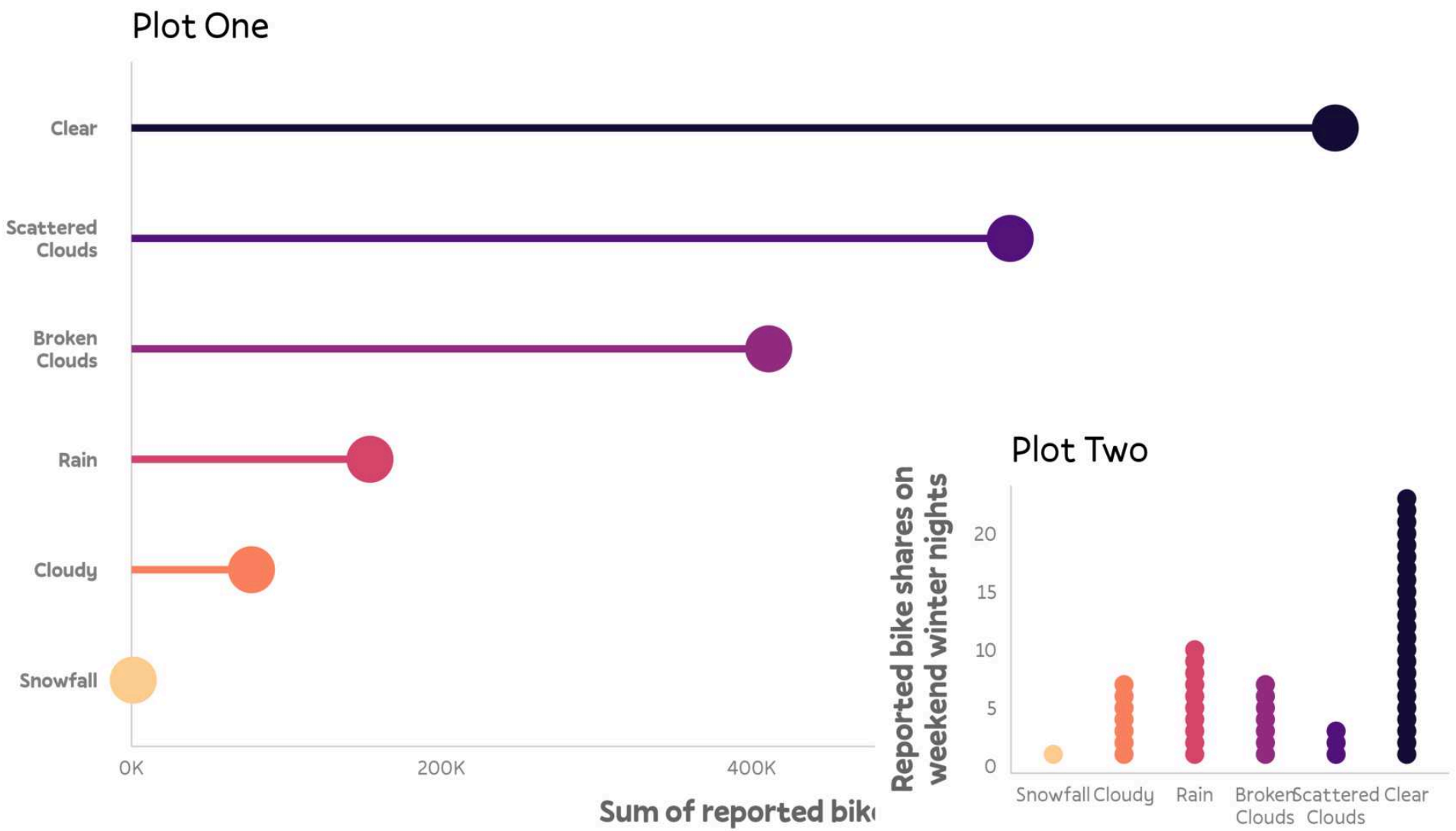
Add Inset Plots

```
1  pl1 + inset_element(pl2, l = .6, b = .1, r = 1, t = .6)
```



Add Inset Plots

```
1  pl1 + inset_element(pl2, l = .6, b = 0, r = 1, t = .5, align_to = 'full')
```

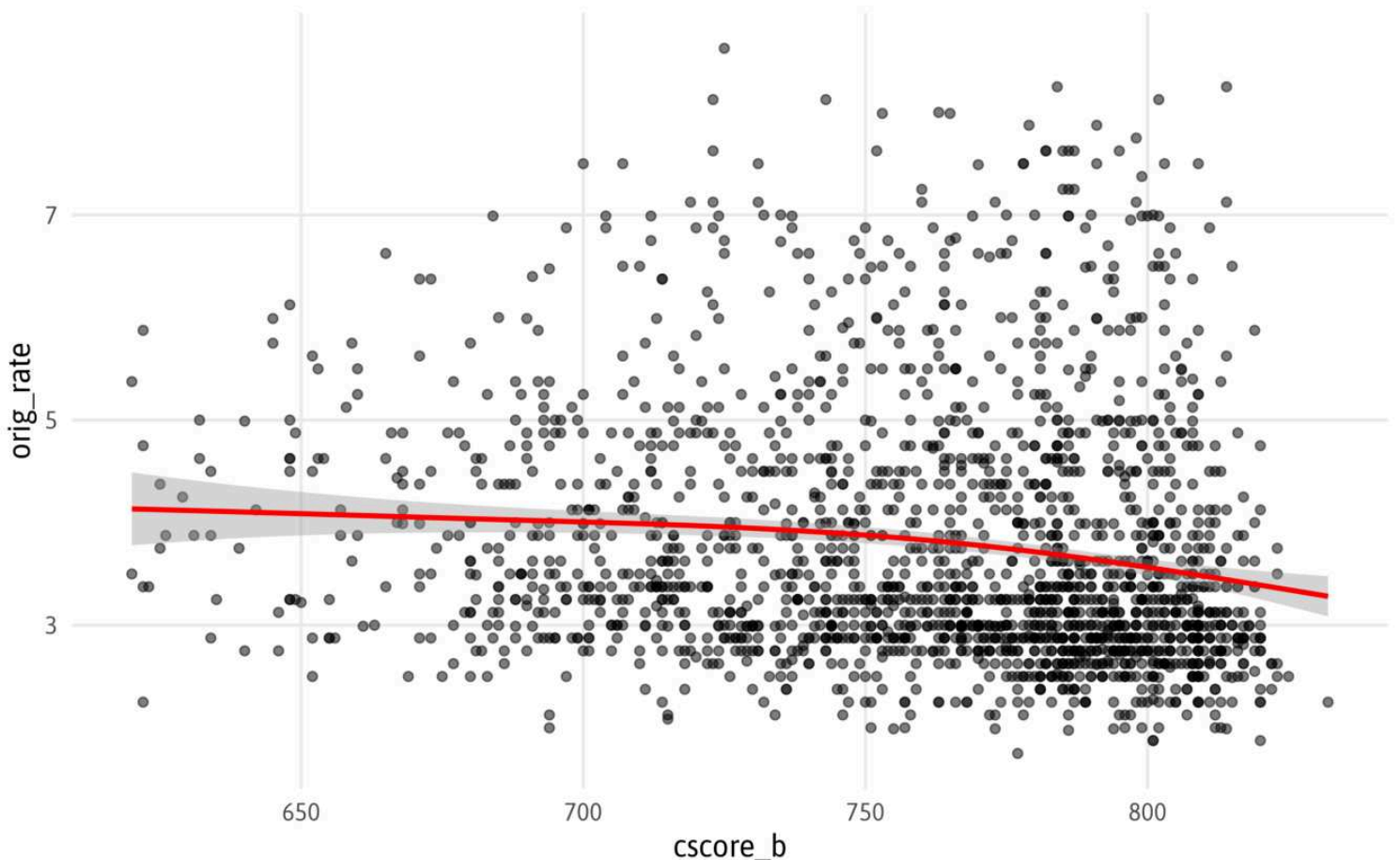


Functional Programming with ggplot2

Cédric Scherer Data Visualization & Information Design

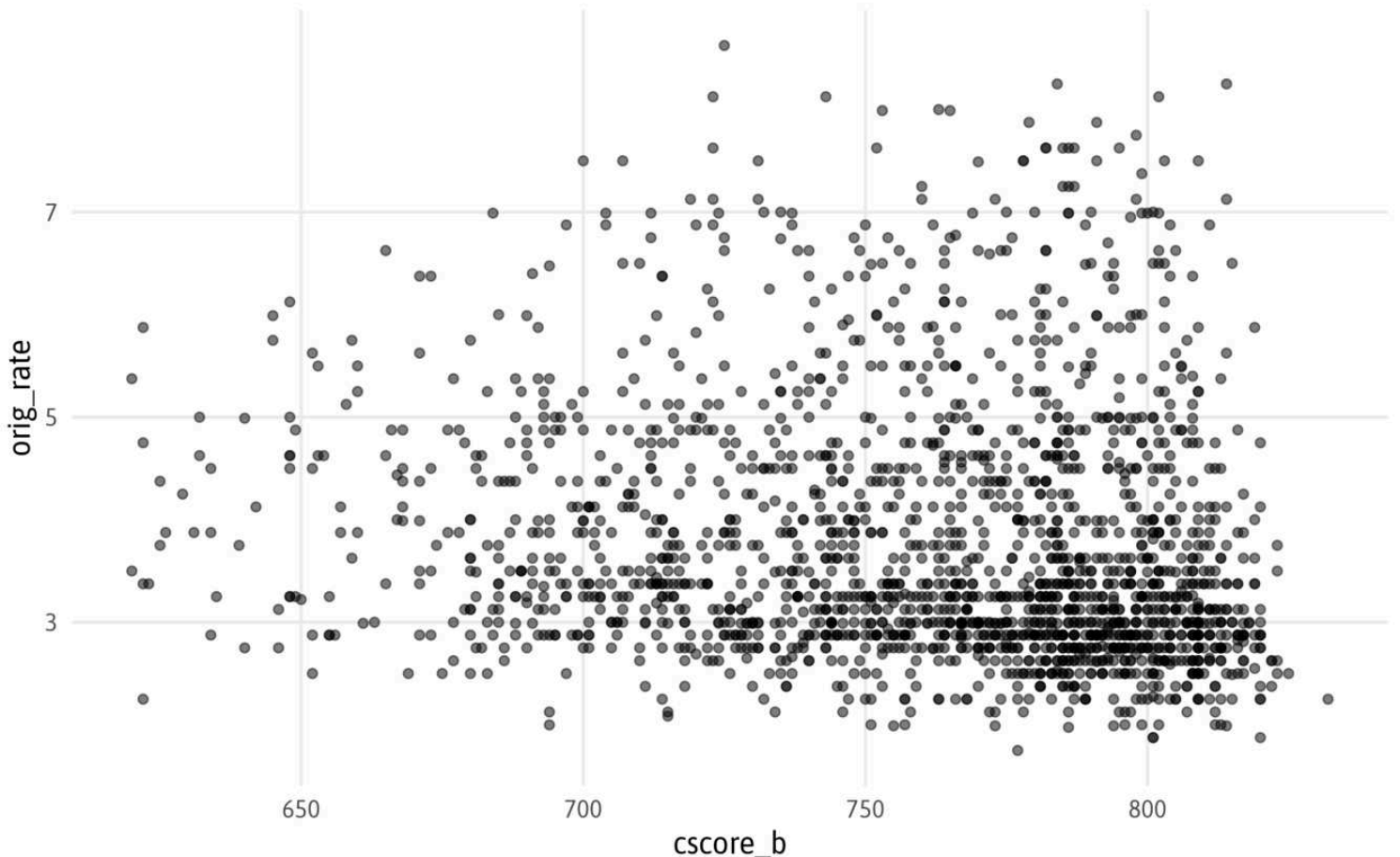
Conditional Components

```
1  smooth <- TRUE
2
3  ggplot(fannie_sub, aes(x = cscore_b, y = orig_rate)) +
4    geom_point(alpha = .5) +
5    { if(smooth) geom_smooth(color = "red") }
```



Conditional Components

```
1  smooth <- FALSE
2
3  ggplot(fannie_sub, aes(x = cscore_b, y = orig_rate)) +
4    geom_point(alpha = .5) +
5    { if(smooth) geom_smooth(color = "red") }
```

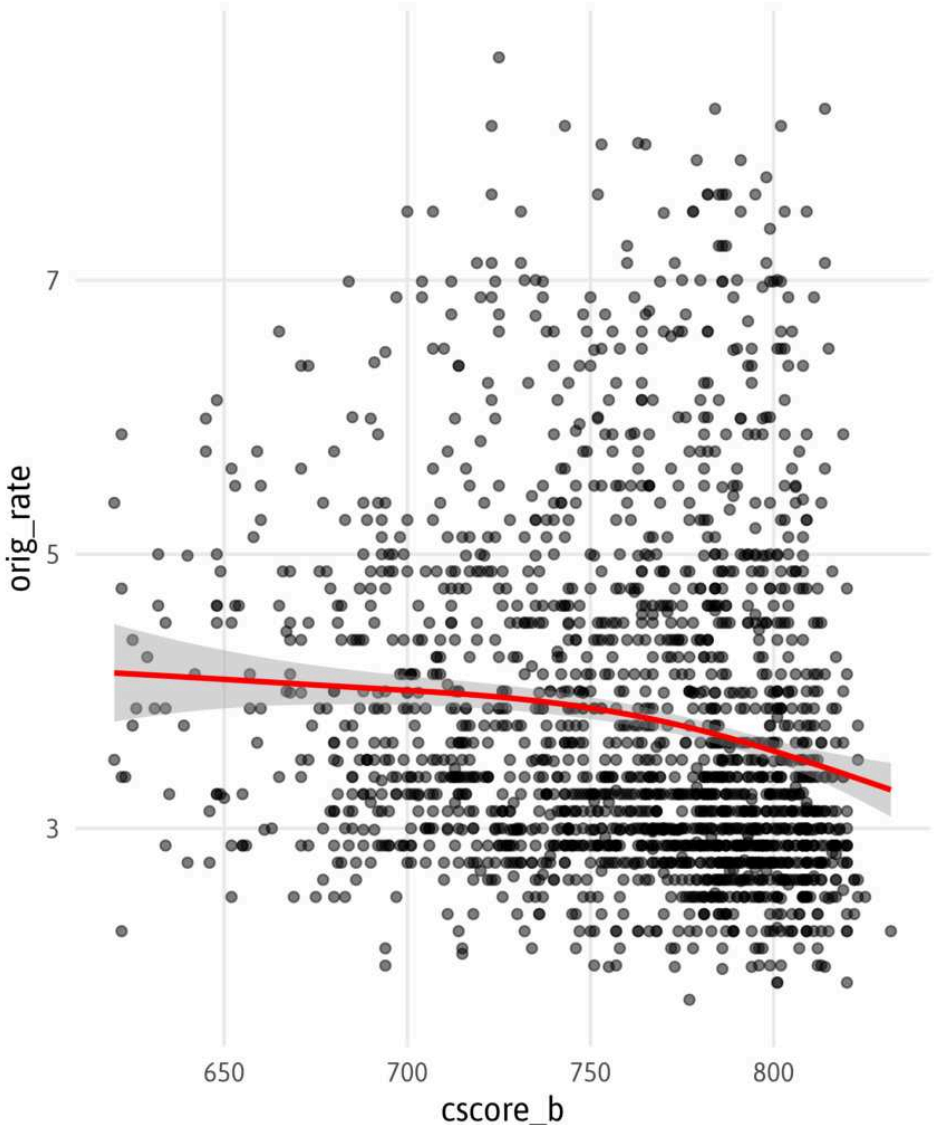


Wrapper Functions for Plots

```
1 draw_scatter <- function(smooth = TRUE) {
2   ggplot(fannie_sub, aes(x = cscore_b, y = orig_rate)) +
3     geom_point(alpha = .5) +
4     { if(smooth) geom_smooth(color = "red") }
5 }
```

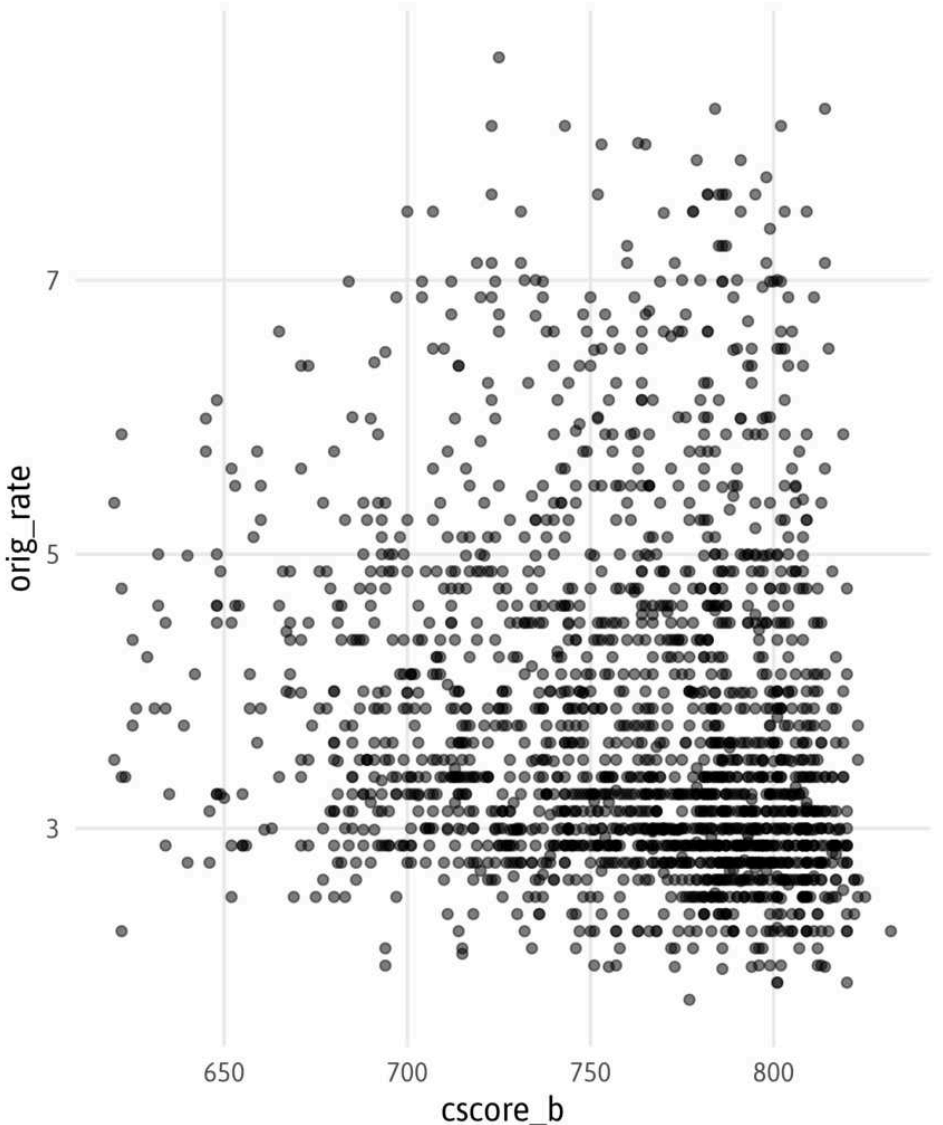
Wrapper Functions for Plots

```
1  draw_scatter()
```



Wrapper Functions for Plots

```
1  draw_scatter(smooth = FALSE)
```

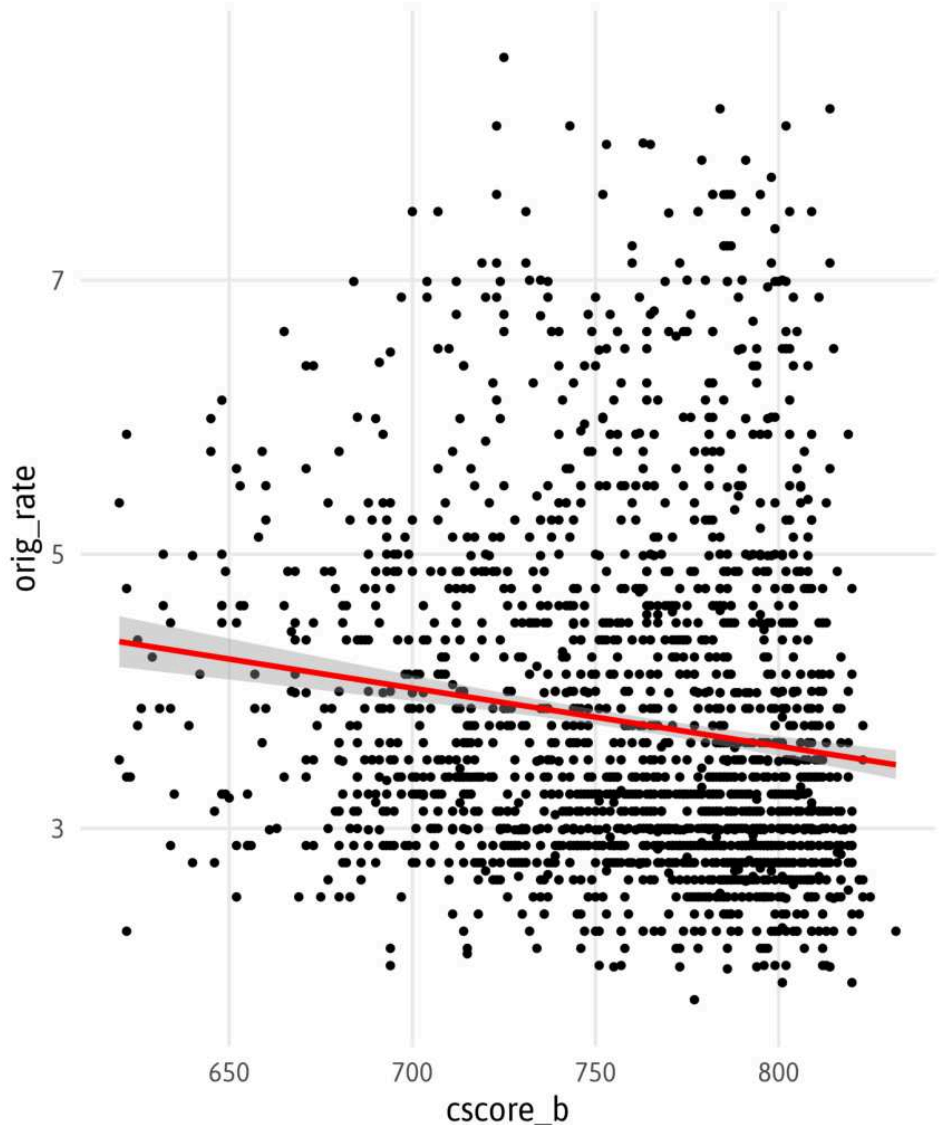


Components as Functions

```
1 geom_scatterfit <- function(pointsize = 1, pointalpha = 1,
2                             method = "lm", linecolor = "red", ...) {
3   list(
4     geom_point(size = pointsize, alpha = pointalpha, ...),
5     geom_smooth(method = method, color = linecolor, ...)
6   )
7 }
```

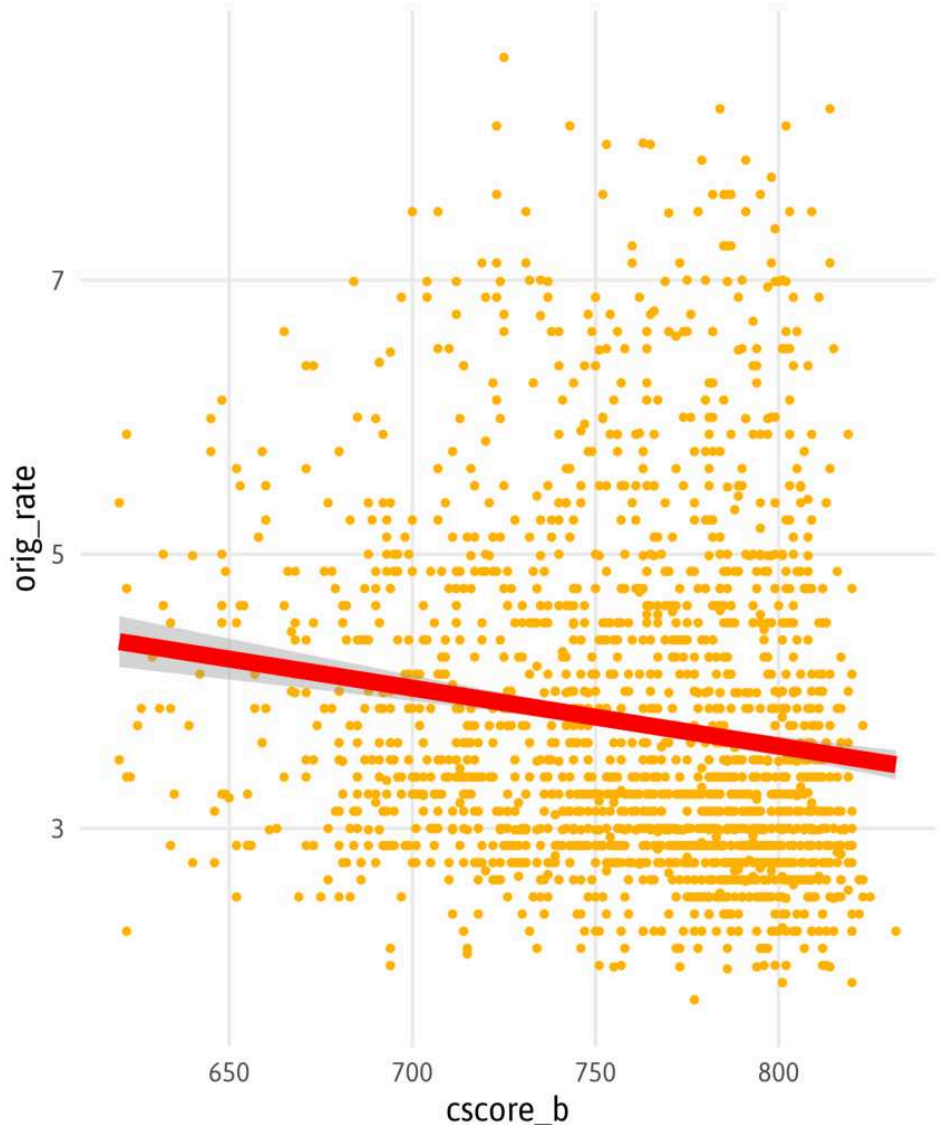
Components as Functions

```
1 ggplot(fannie_sub,
2        aes(x = cscore_b, y = orig_r
3   geom_scatterfit()
```



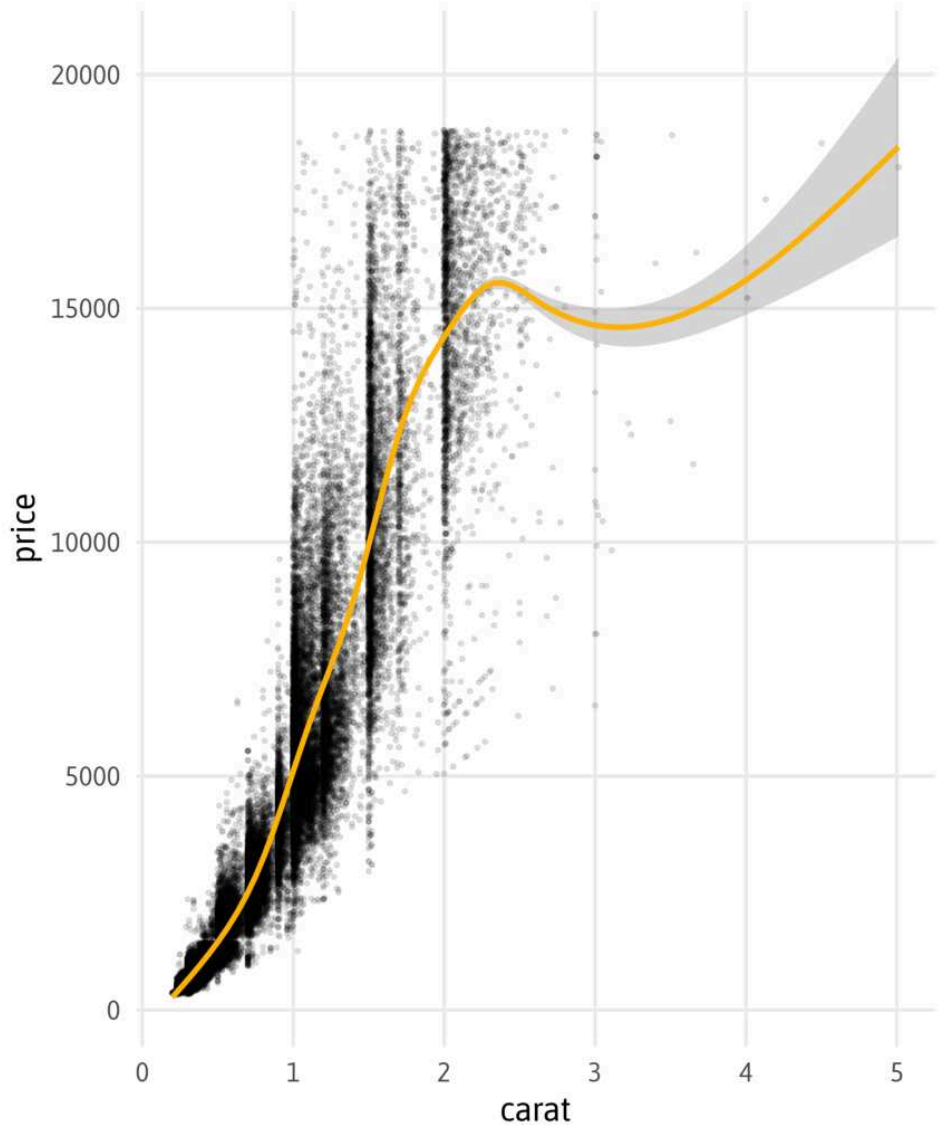
Components as Functions

```
1 ggplot(fannie_sub,
2        aes(x = cscore_b, y = orig_r
3   geom_scatterfit(
4     color = "#FFB306",
5     linewidth = 3
6   )
```



Components as Functions

```
ggplot(diamonds,
       aes(x = carat, y = price)) +
  geom_scatterfit(
    pointsize = .5,
    pointalpha = .1,
    method = "gam",
    linecolor = "#FFB306"
  )
```

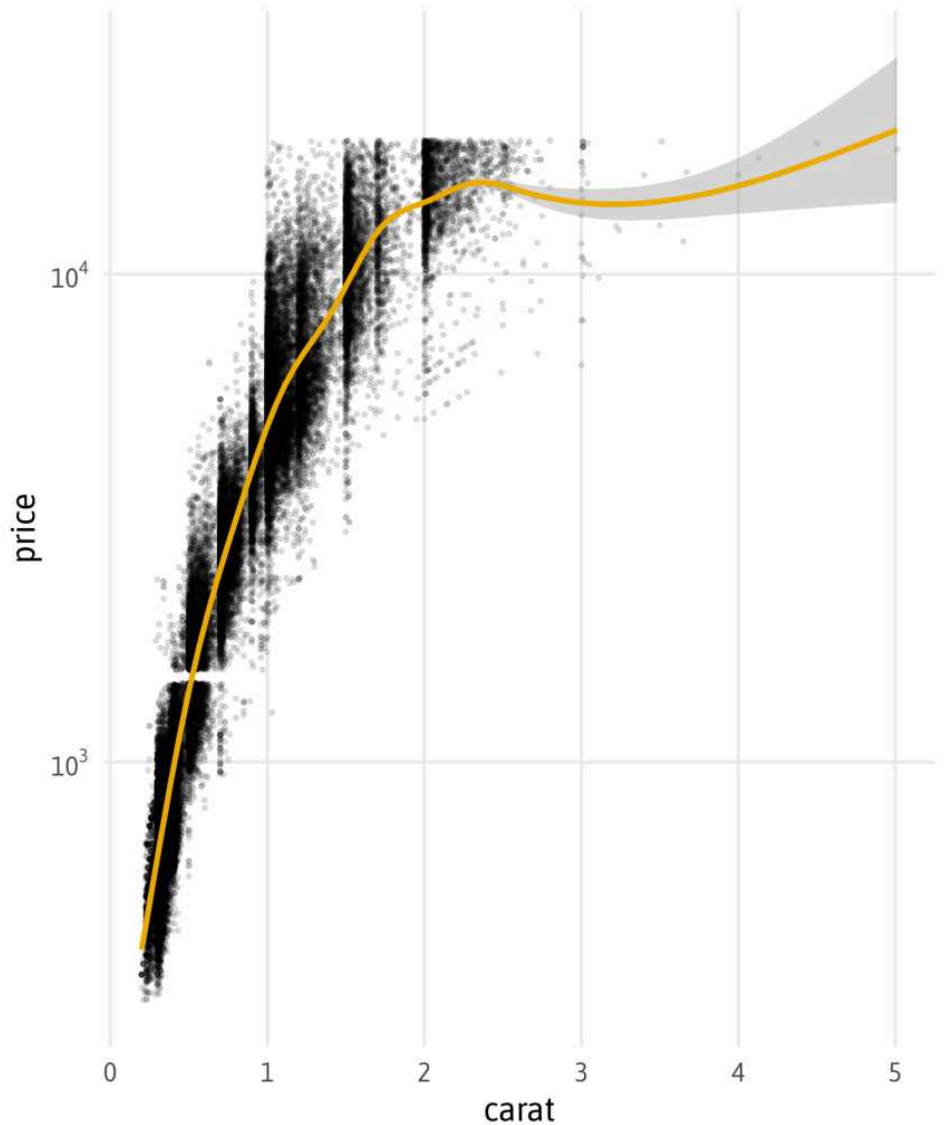


Components as Functions

```r
scales_log <- function(sides = "xy") {
  list(
    if(stringr::str_detect(sides, "x")) {
      scale_x_log10(
        breaks = c(10^(1:100)), labels = scales::label_log()
      )
    },
    if(stringr::str_detect(sides, "y")) {
      scale_y_log10(
        breaks = c(10^(1:100)), labels = scales::label_log()
      )
    }
  )
}
```

Components as Functions

```
1 ggplot(diamonds,
2        aes(x = carat, y = price)) +
3   geom_scatterfit(
4     pointsize = .5,
5     pointalpha = .1,
6     method = "gam",
7     linecolor = "#EFAC00"
8   ) +
9   scales_log(sides = "y")
```

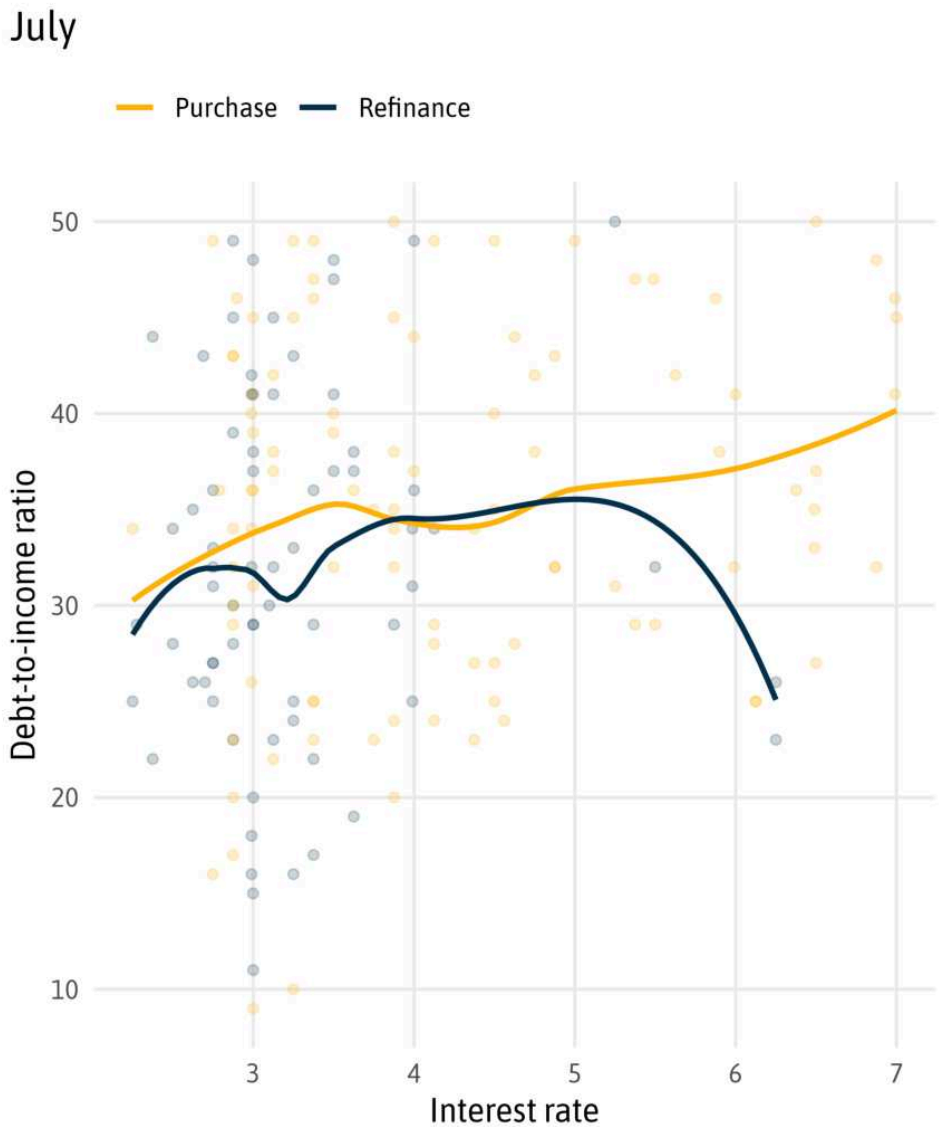


Iterative Graphics

```
trends_monthly <- function(grp = "January") {
  fannie_sub |>
    mutate(month = lubridate::month(orig_date, label = TRUE, abbr = FALSE)) |>
    filter(month %in% grp) |>
    ggplot(aes(x = orig_rate, y = dti, color = purpose)) +
    geom_point(alpha = .2, show.legend = FALSE) +
    geom_smooth(se = FALSE) +
    scale_color_manual(values = c("#FFB306", "#05314D")) +
    labs(title = grp, x = "Interest rate", y = "Debt-to-income ratio", color = NULL)
}
```

Iterative Graphics

```
1  trends_monthly("July")
```

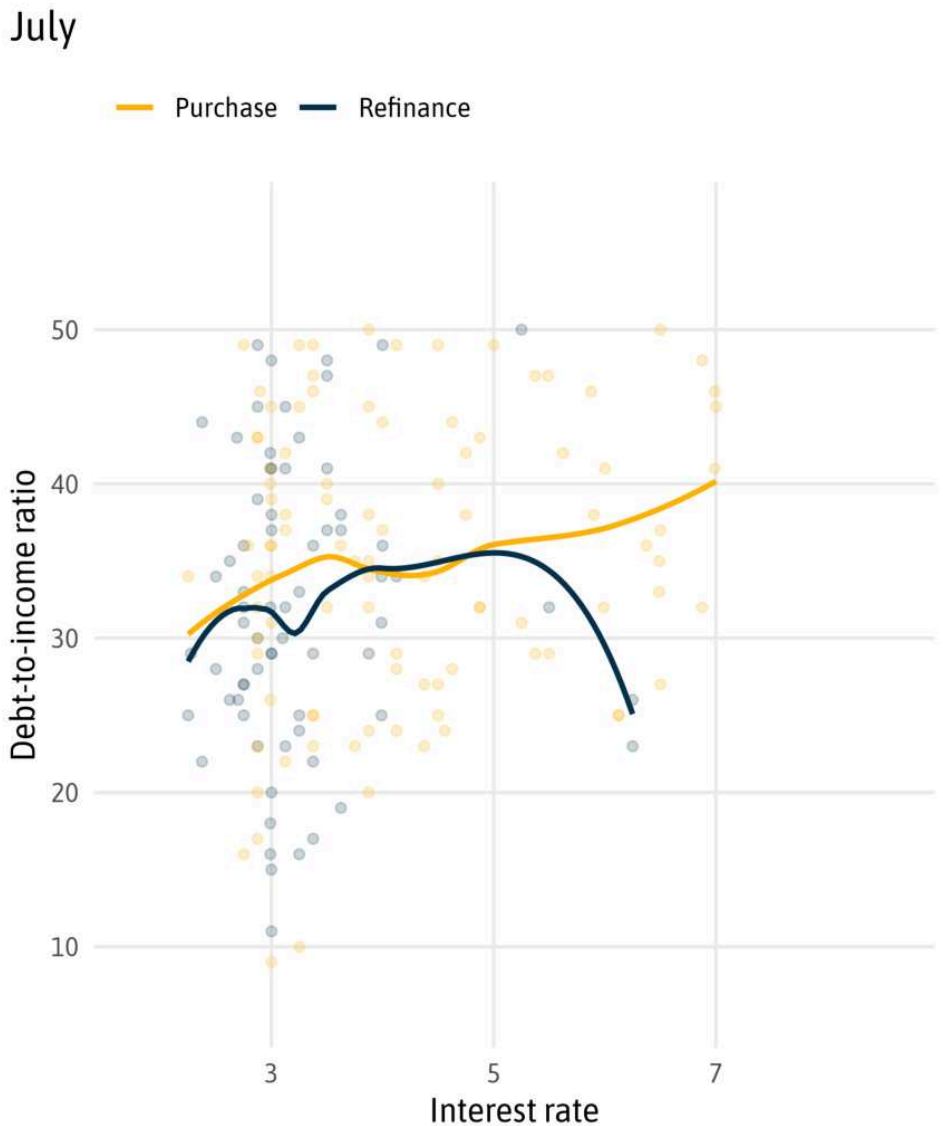


Iterative Graphics

```
trends_monthly <- function(grp = "January") {
  fannie_sub |>
    mutate(month = lubridate::month(orig_date, label = TRUE, abbr = FALSE)) |>
    filter(month %in% grp) |>
    ggplot(aes(x = orig_rate, y = dti, color = purpose)) +
    geom_point(alpha = .2, show.legend = FALSE) +
    geom_smooth(se = FALSE) +
    # keep axis ranges consistent
    scale_x_continuous(limits = range(fannie_sub$orig_rate)) +
    scale_y_continuous(limits = range(fannie_sub$dti)) +
    scale_color_manual(values = c("#FFB306", "#05314D")) +
    labs(title = grp, x = "Interest rate", y = "Debt-to-income ratio", color = NULL)
}
```

Iterative Graphics

```
1  trends_monthly("July")
```

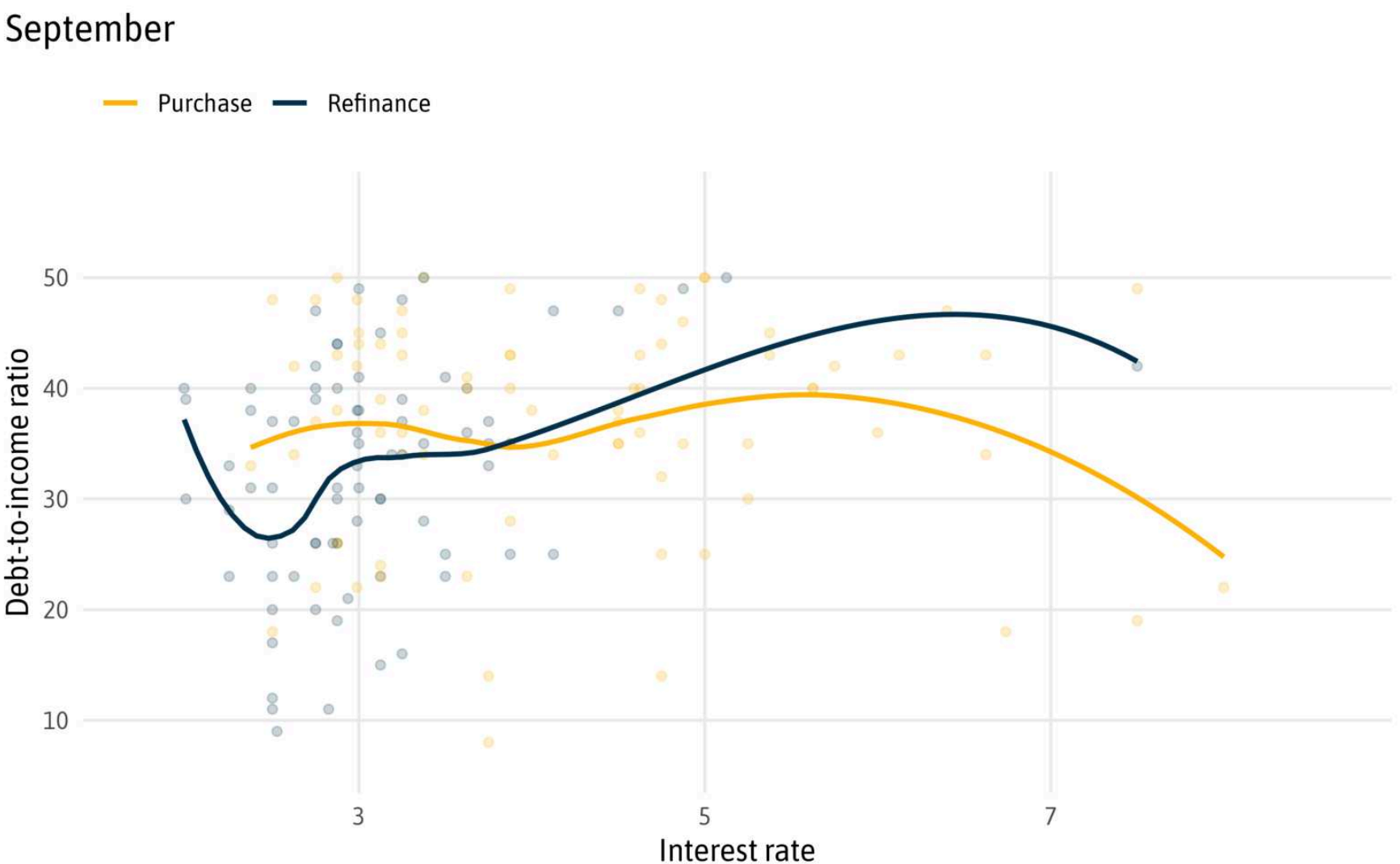


Iterative Graphics

```
plots <- purrr::map(month.name[1:12], trends_monthly) ## also: ~ trends_monthly(.x)
```

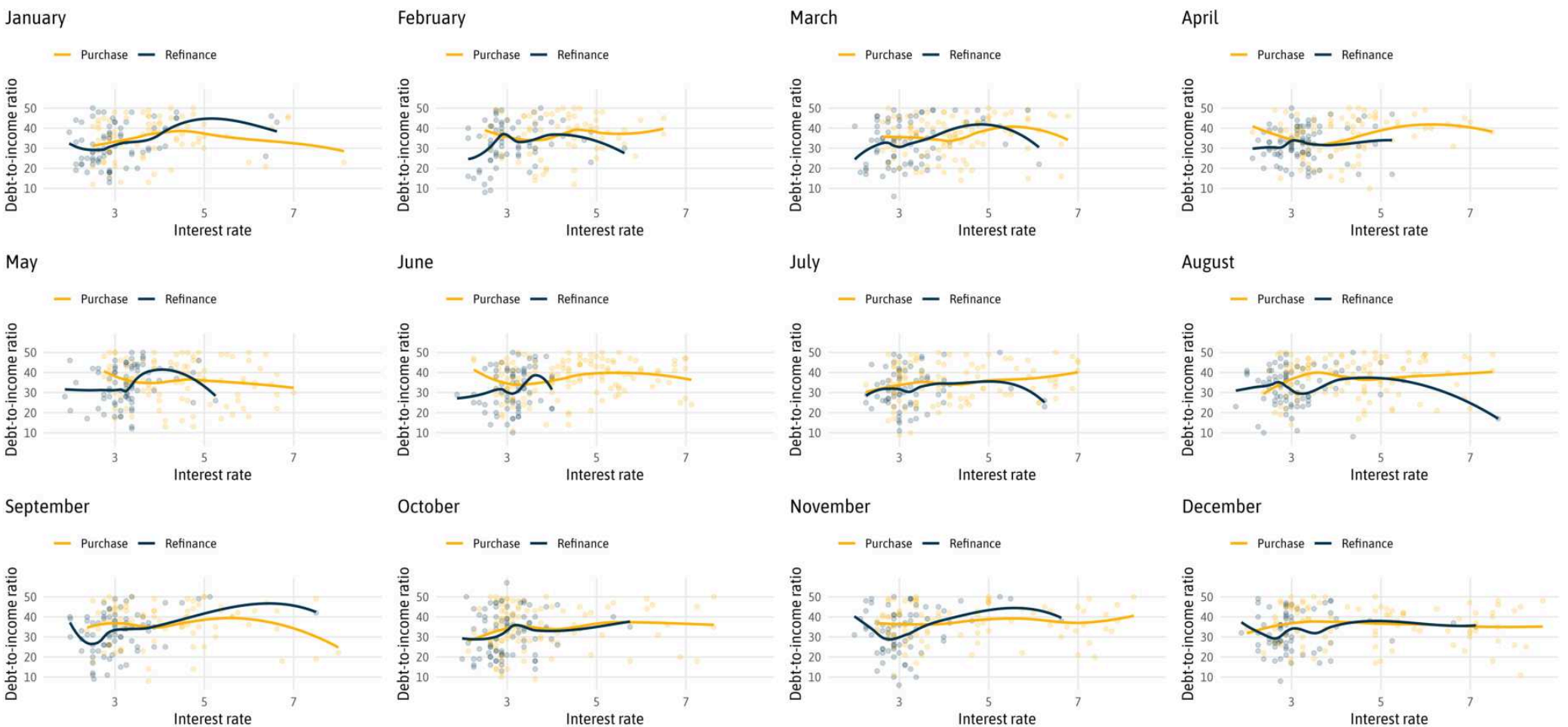
Iterative Graphics

```
1 plots <- purrr::map(month.name[1:12], trends_monthly) ## also: ~ trends_monthly(.x)
2 plots[[9]]
```



Iterative Graphics

```
1 plots <- purrr::map(month.name[1:12], trends_monthly) ## also: ~ trends_monthly(.x)
2 patchwork::wrap_plots(plots)
```

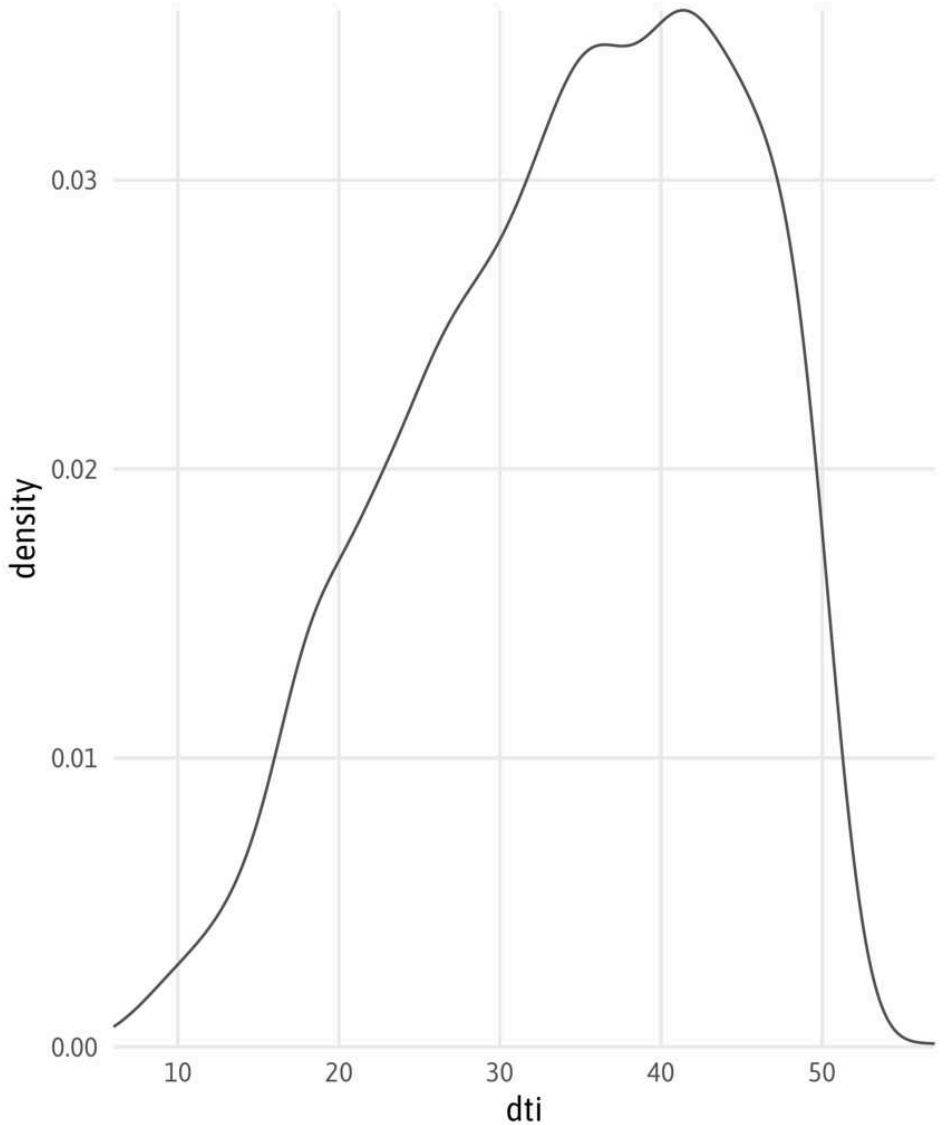


Iterative Graphics

```r
plot_density <- function(data, var, grp = "") {
  ggplot(data, aes(x = !!sym(var))) +
    geom_density(aes(fill = !!sym(grp)), position = "identity",
                 color = "grey30", alpha = .3) +
    coord_cartesian(expand = FALSE, clip = "off") +
    scale_y_continuous(labels = scales::label_number()) +
    scale_fill_brewer(palette = "Dark2", name = NULL) +
    theme(legend.position = "top")
}
```

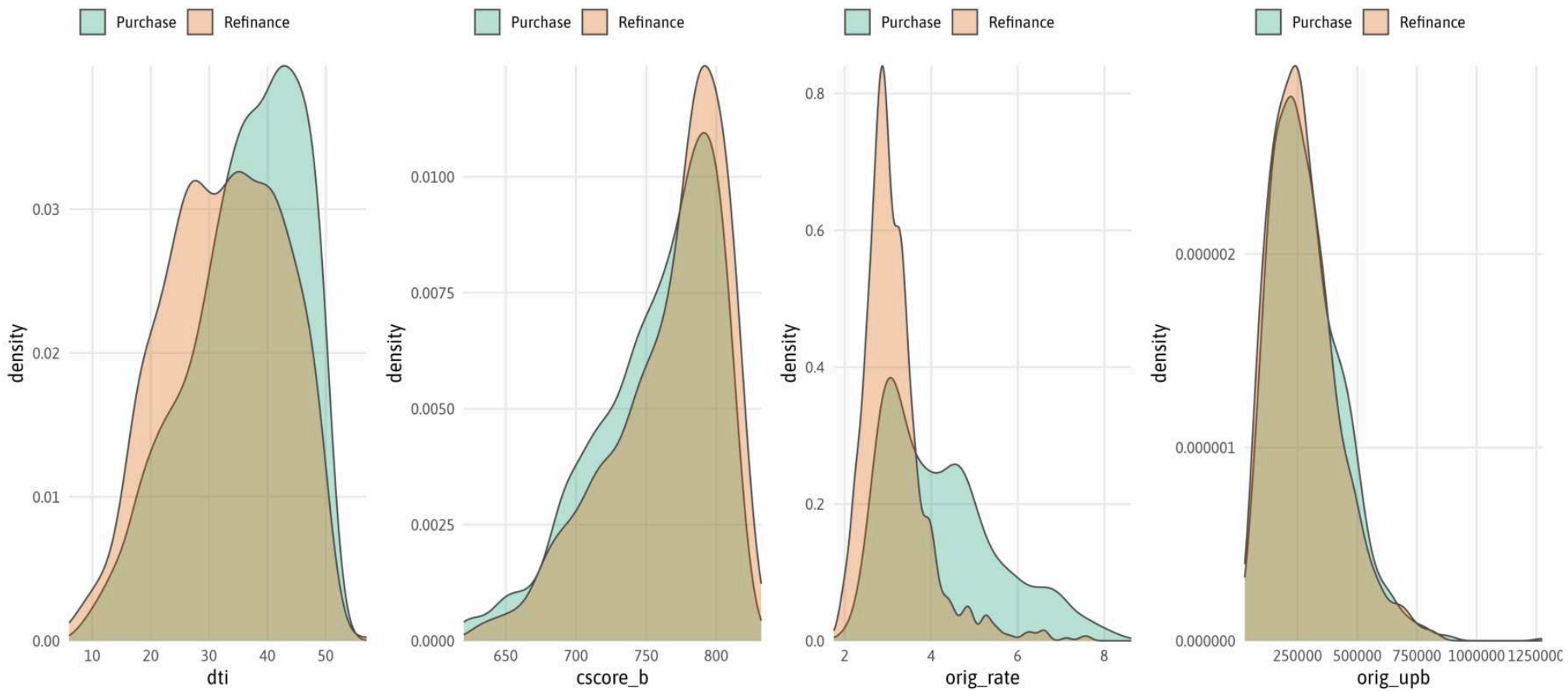
Iterative Graphics

```
1 plot_density(
2   fannie_sub, "dti"
3 )
```



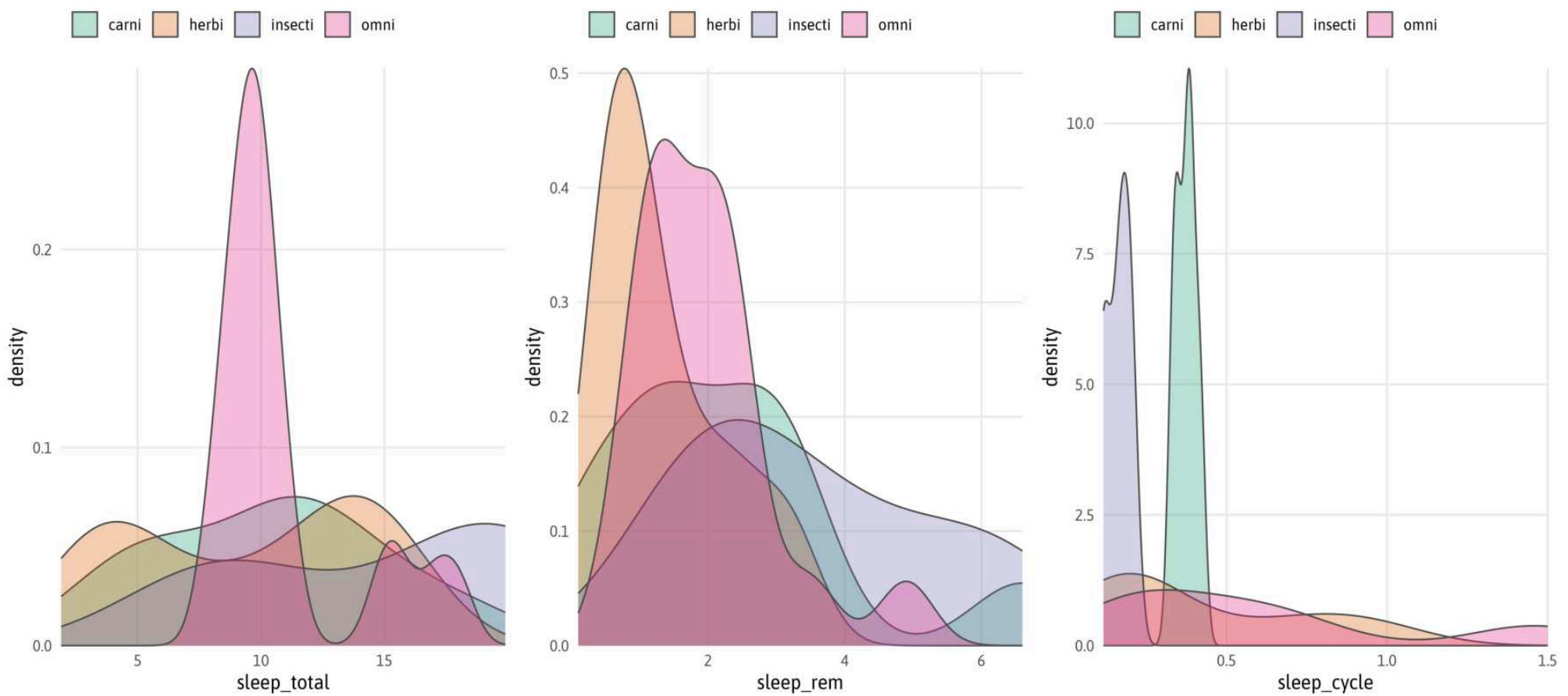
Iterative Graphics

```
plots <- purrr::map(
  c("dti", "cscore_b", "orig_rate", "orig_upb"),
  ~ plot_density(data = fannie_sub, var = .x, grp = "purpose")
)
patchwork::wrap_plots(plots, nrow = 1)
```



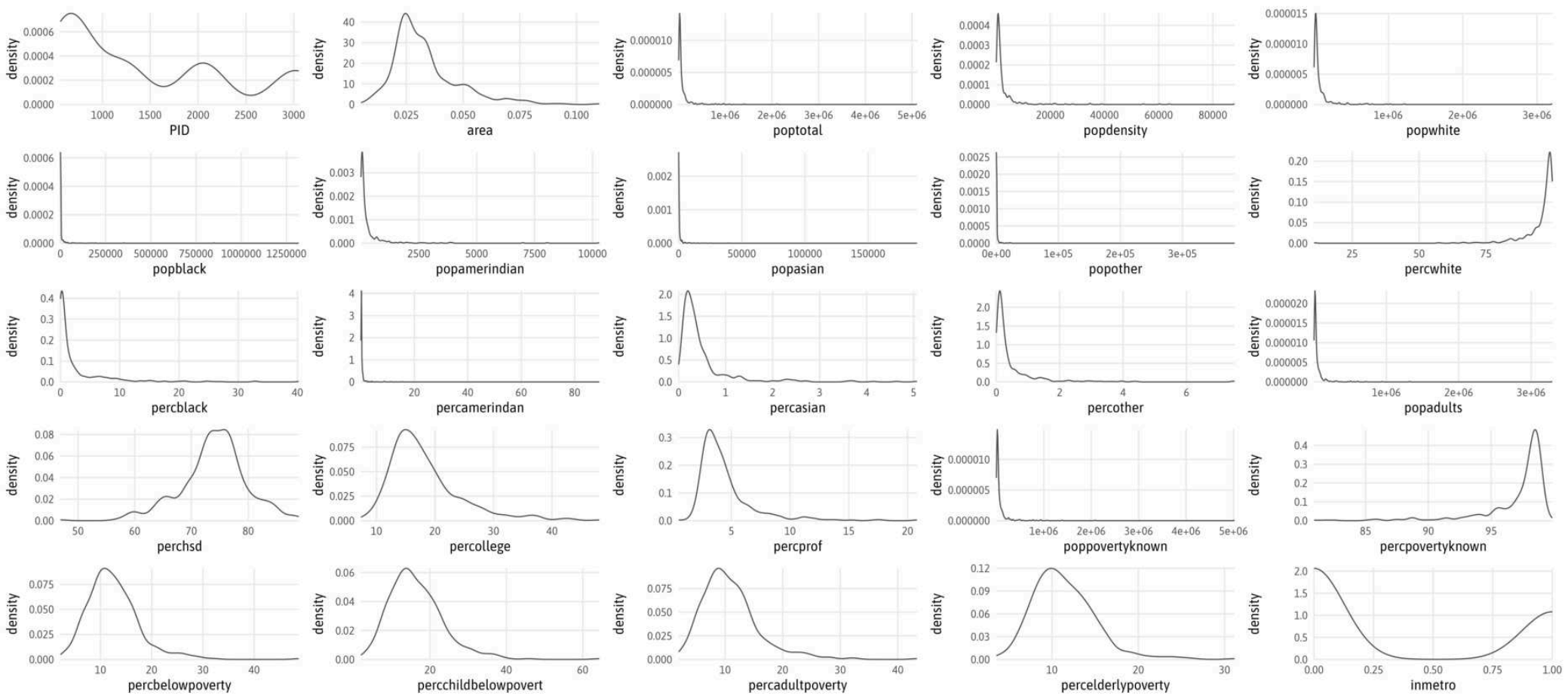
Iterative Graphics

```
plots <- purrr::map(
  c("sleep_total", "sleep_rem", "sleep_cycle"),
  ~ plot_density(data = filter(msleep, !is.na(vore)), var = .x, grp = "vore")
)
patchwork::wrap_plots(plots, nrow = 1)
```



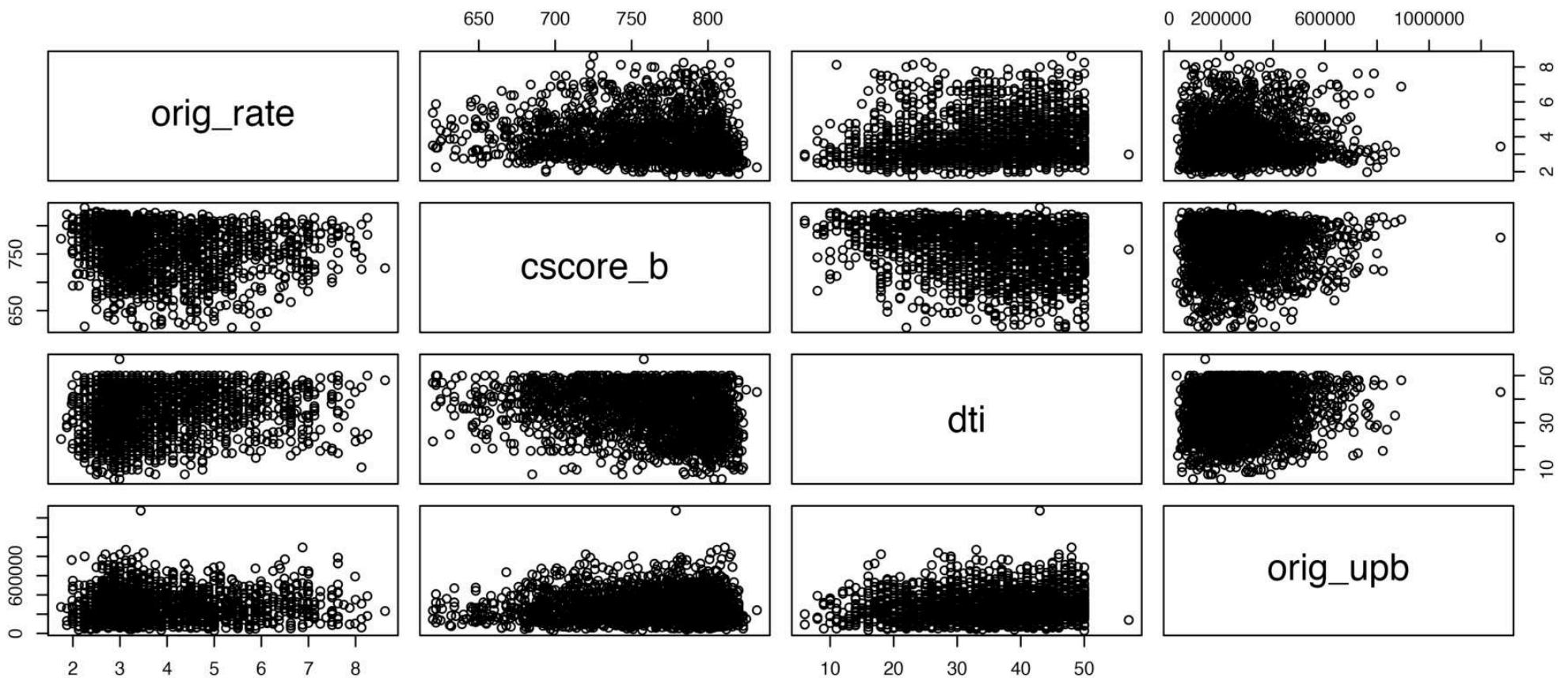
Iterative Graphics

```
plots <- purrr::map(
  names(select(midwest, where(is.numeric))),
  ~plot_density(data = midwest, var = .x)
)
patchwork::wrap_plots(plots)
```



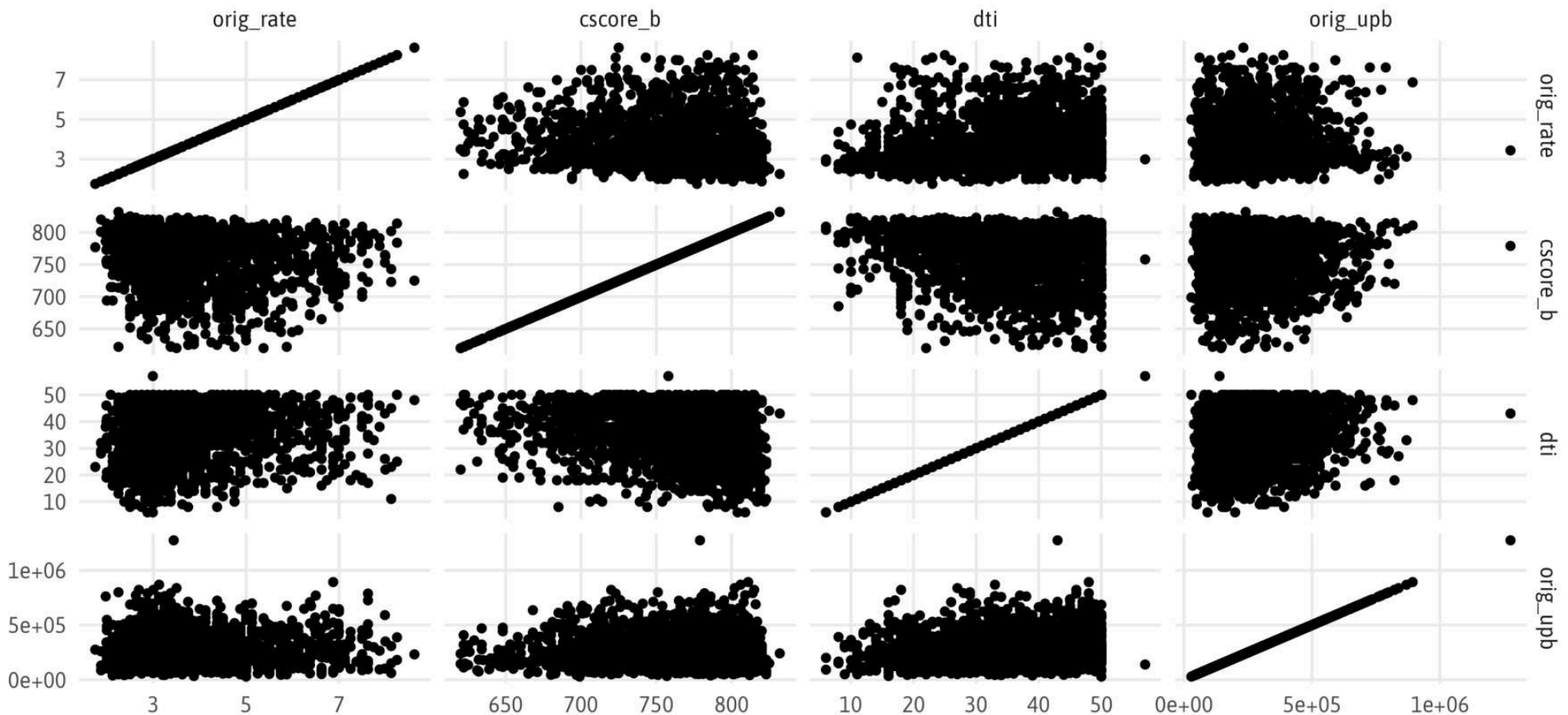
Create Grids of Variables

```
1  plot(fannie_sub[, 2:5])
```



Create Grids of Variables

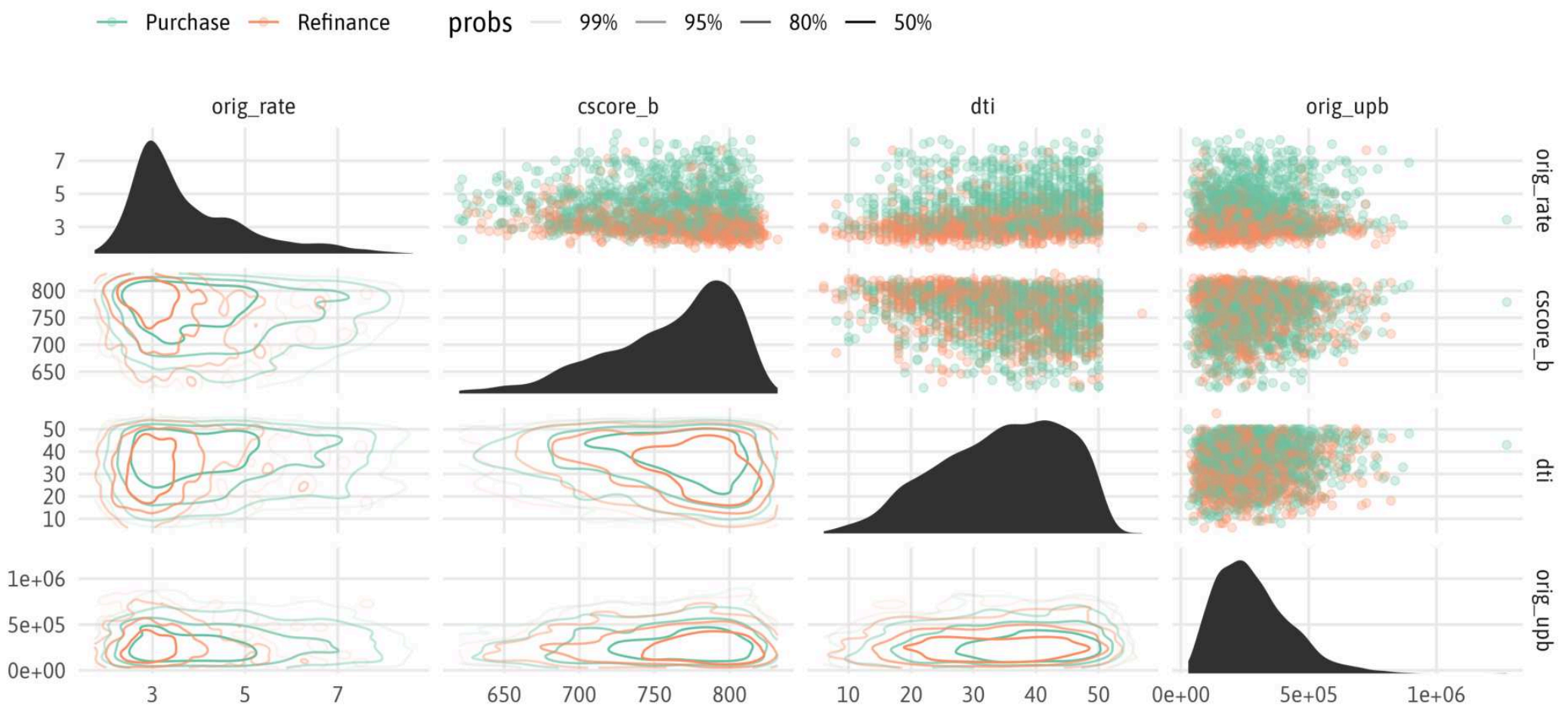
```
1 ggplot(fannie_sub, aes(x = .panel_x, y = .panel_y)) +
2   geom_point() +
3   ggforce::facet_matrix(
4     vars(orig_rate, cscore_b, dti, orig_upb)
5   )
```



Create Grids of Variables

```
ggplot(fannie_sub, aes(x = .panel_x, y = .panel_y)) +
  geom_point(aes(color = purpose), alpha = .25) +
  ggdensity::geom_hdr_lines(aes(color = purpose), linewidth = .5) +
  ggforce::geom_autodensity() +
  scale_color_brewer(palette = "Set2", name = NULL) +
  ggforce::facet_matrix(
    vars(orig_rate, cscore_b, dti, orig_upb),
    layer.lower = 2, layer.diag = 3
  )
```

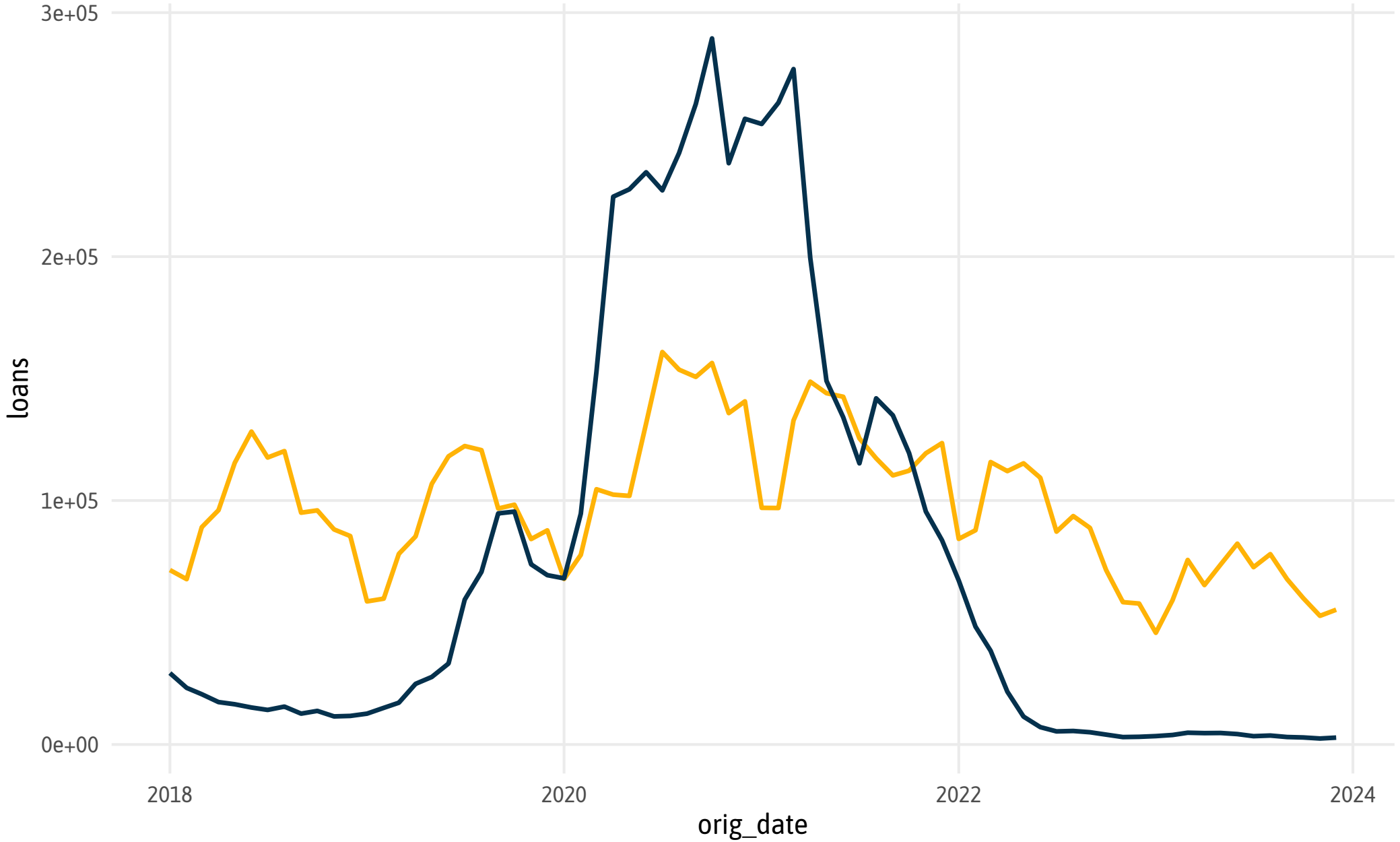
Create Grids of Variables



Interactive Graphics with {ggplot2}

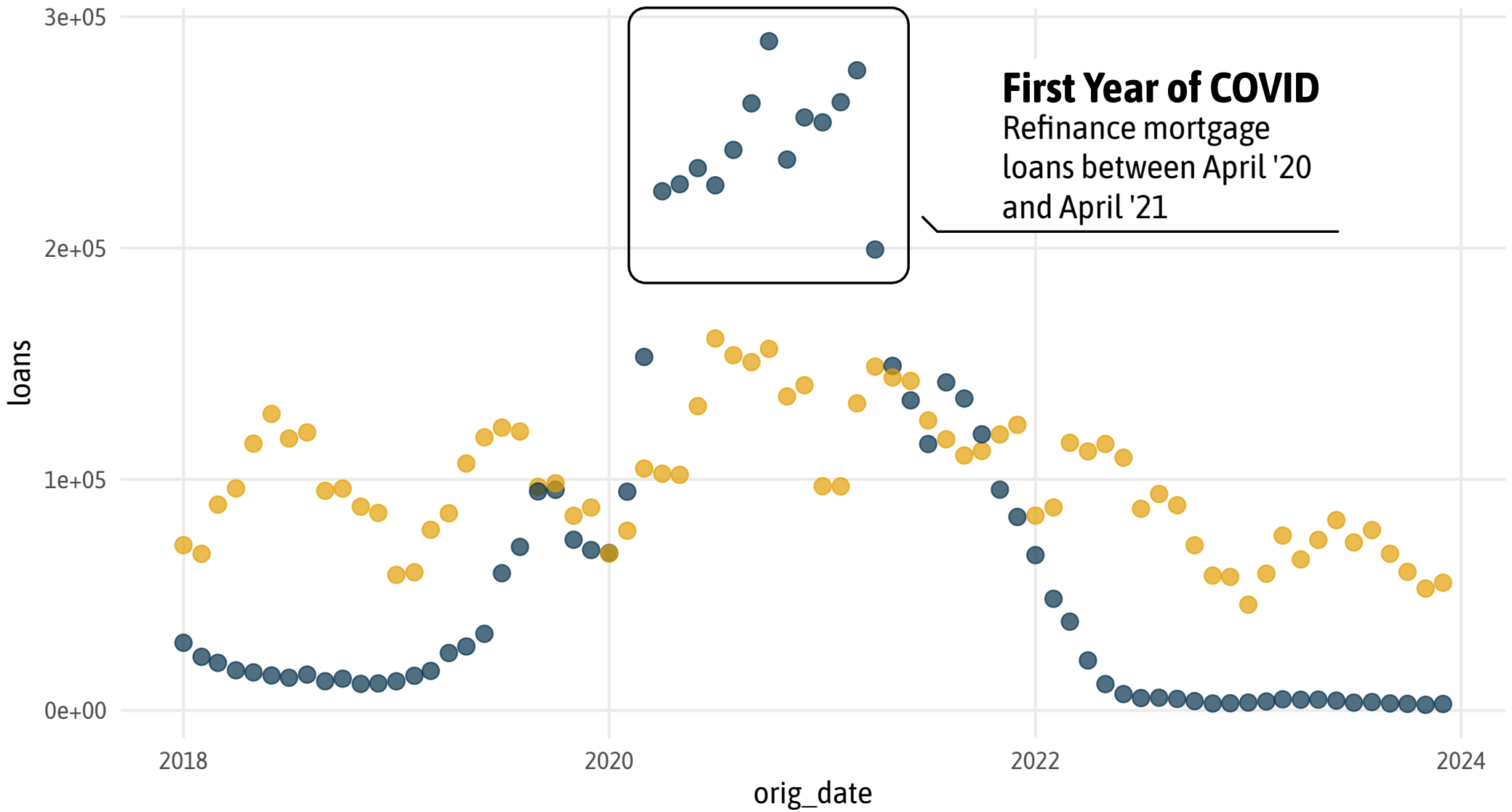
Cédric Scherer Data Visualization & Information Design

```r
p1 <-
  ggplot(fannie_agg, aes(x = orig_date, y = loans, color = purpose, group = purpose)) +
  ggiraph::geom_line_interactive(aes(tooltip = purpose, data_id = purpose), linewidth = 1) +
  scale_color_manual(values = c("#FFB306", "#05314D"), guide = "none")

ggiraph::set_girafe_defaults(
  opts_zoom = ggiraph::opts_zoom(min = 1, max = 4),
  opts_toolbar = ggiraph::opts_toolbar(position = "bottomright")
)

ggiraph::girafe(
  ggobj = p1, width_svg = 9, height_svg = 5.5,
  options = list(
    ggiraph::opts_hover_inv(css = "opacity:0.3;"),
    ggiraph::opts_hover(css = "stroke-width:5;")
  )
)
```


3e+05
2e+05
1e+05
0e+00
loans
2018
2020
2022
2024
orig_date

```r
p2 <-
  ggplot(fannie_agg, aes(x = orig_date, y = loans, color = purpose)) +
  ggiraph::geom_point_interactive(aes(tooltip = paste("<b style='font-size:12pt;'>", s
  ggforce::geom_mark_rect(
    aes(label = "First Year of COVID", filter = loans > 190000),
    description = "Refinance mortgage loans between April '20 and April '21",
    color = "black", label.family = "Asap SemiCondensed", label.fontsize = c(18, 14)
  ) +
  coord_cartesian(clip = "off") +
  scale_color_manual(values = c("#E39F03", "#05314D"), guide = "none") +
  ggtitle("<b style='color:#FFB306;'>Purchase money mortgages</b> versus <b style='col
  theme(plot.title = ggtext::element_markdown(size = 20, margin = margin(b = 25)))

ggiraph::girafe(
  ggobj = p2, width_svg = 9, height_svg = 5.5,
  options = list(
    ggiraph::opts_tooltip(use_fill = TRUE, css = "font-size:9pt;color:white;padding:7p
    ggiraph::opts_hover(css = "stroke-width:8px;opacity:1;"),
    ggiraph::opts_hover_inv(css = "opacity:0.3;")
  )
)
```

Purchase money mortgages versus refinance mortgages



Exciting Extensions

Cédric Scherer Data Visualization & Information Design

Layers

- **{geofacet}** – tile grid maps
- **{ggalluvial}** – alluvial plots
- **{ggalt}** – dumbbell, horizon, and lollipop charts, splines, ...
- **{ggbeeswarm}** – beeswarm plots and variants
- **{ggbraid}** – ribbons for alternating groups
- **{ggbump}** – bump charts
- **{ggdensity}** – improved density plots
- **{ggdist}** – uncertainty visualizations
- **{ggforce}** – several interesting layers (and more)
- **{ggpattern}** – pattern fills for layers

Layers (continued)

- `{ggpointdensity}` – density gradients for scatter plots
- `{ggraph}` – networks, graphs & trees
- `{ggridges}` – ridgeline plots
- `{ggsankey}` – sankey diagrams
- `{ggsignif}` – significance levels
- `{ggstar}` – more point shapes
- `{ggstream}` – stream graphs
- `{ggupset}` – upset graphs
- `{treemapify}` – treemaps

Utilities

- **{cowplot}** – combine ggplots
- **{ggannotate}** – point-n-click annotations
- **{ggblend}** – blend, compose, adjust layers
- **{ggfittext}** – scale text according to space
- **{ggfx}** – shaders and filters for layers
- **{ggh4x}** – facets, positions, and more
- **{ggtext}** – text rendering for theme elements + text layers
- **{lemon}** – axis lines (and a few layers)
- **{marquee}** – advanced Markdown rendering
- **{patchwork}** – combine ggplots
- **{scales}** – control scales

Themes

- `{ggdark}`
- `{ggsci}` (also color scales)
- `{ggtech}` (also color scales)
- `{ggthemes}` (also color scales)
- `{ggthemr}`
- `{hrbrthemes}` (also color scales)
- `{tvthemes}` (also color scales)

Color Palettes

- {colorspace}
- {jcolors}
- {MetBrewer}
- {nord}
- {rcartocolor}
- {RColorBrewer}
- {scico}
- {unikn}
- {viridis}
- {wesanderson}

Interactive Charts

- {ggiraph}
- {plotly}
- {echarts4r}*
- {highcharter}*
- {charter}*
- {streamgraph}*
- {tmap}*
- {leaflet}*
- {globe4r}*
- {grapher}*

* not using ggplot2

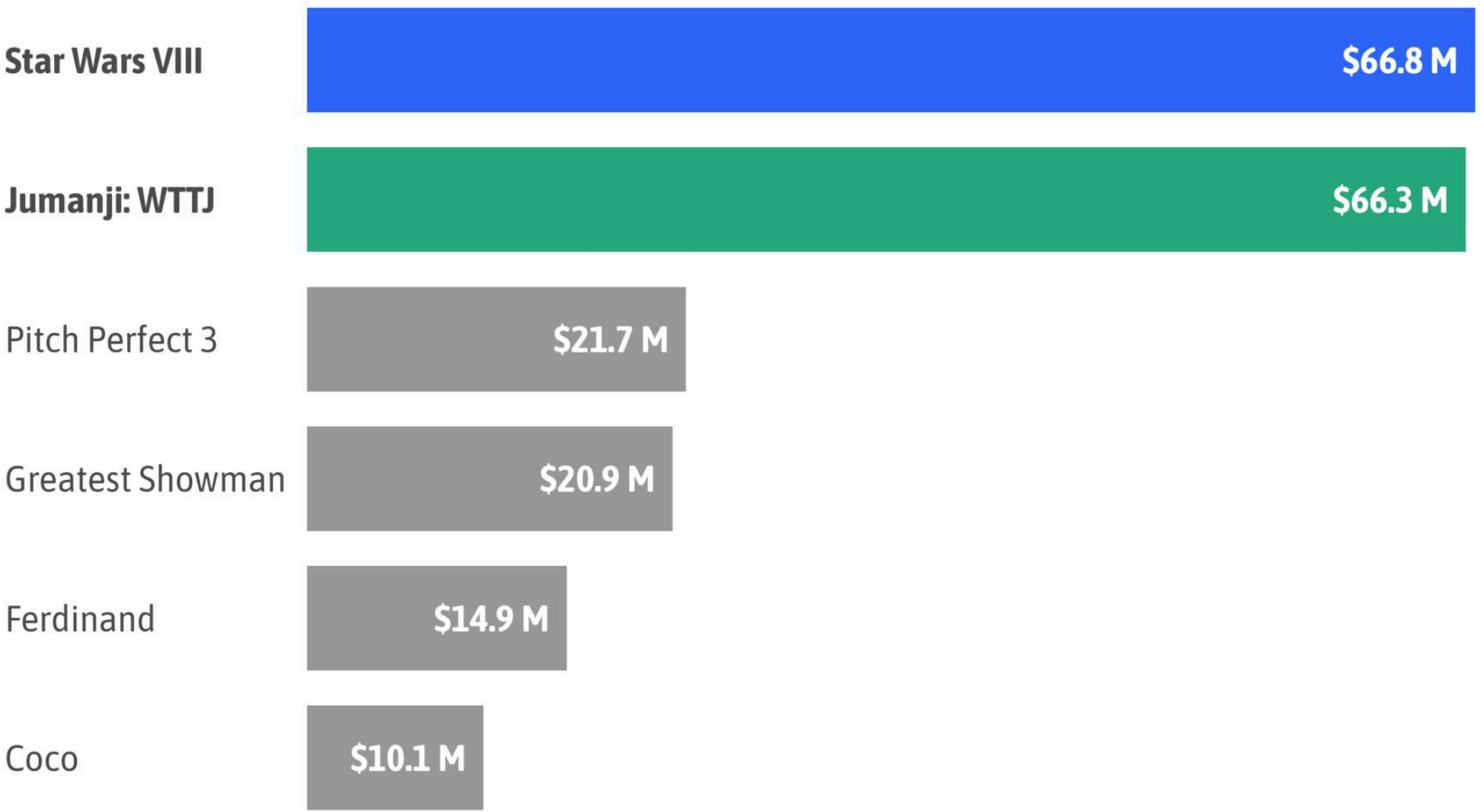
Exercises

Cédric Scherer Data Visualization & Information Design

Exercise 1

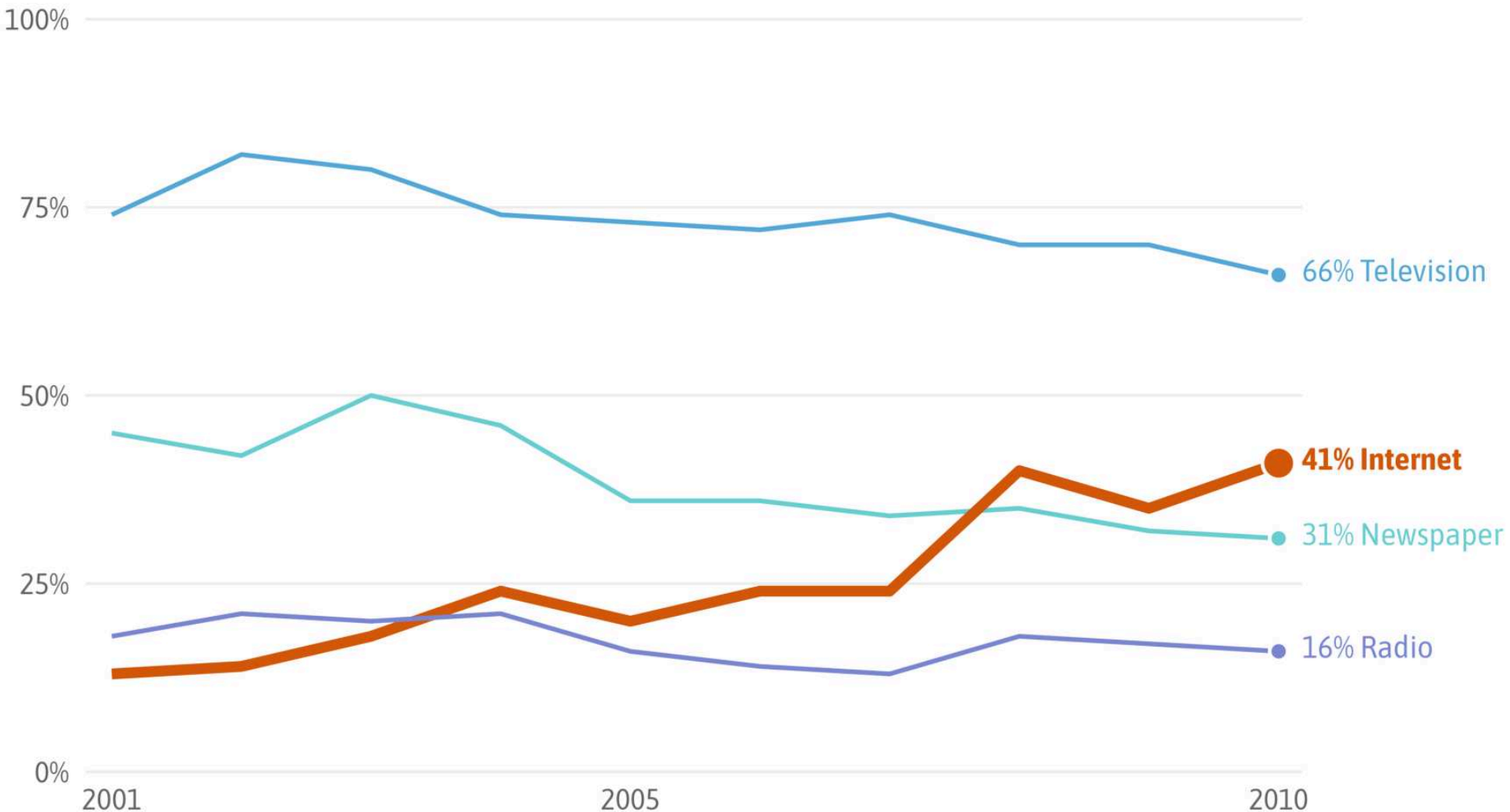
- **Create one of the following visualizations as close as possible:**
  - A horizontal bar chart of box offices, highlighting the top movies “Star Wars VIII” and “Jumanji: Welcome to the Jungle”.
  - A line chart on where Americans get most of their news from 2001 to 2010, highlighting the increase of internet as primary news source.

“Stars Wars VIII – The Last Jedi” just barely topped the box office this weel facing stiff competition from “Jumanji: Welcome to the Jungle”.



An increasing proportion cite the internet as their primary news source.

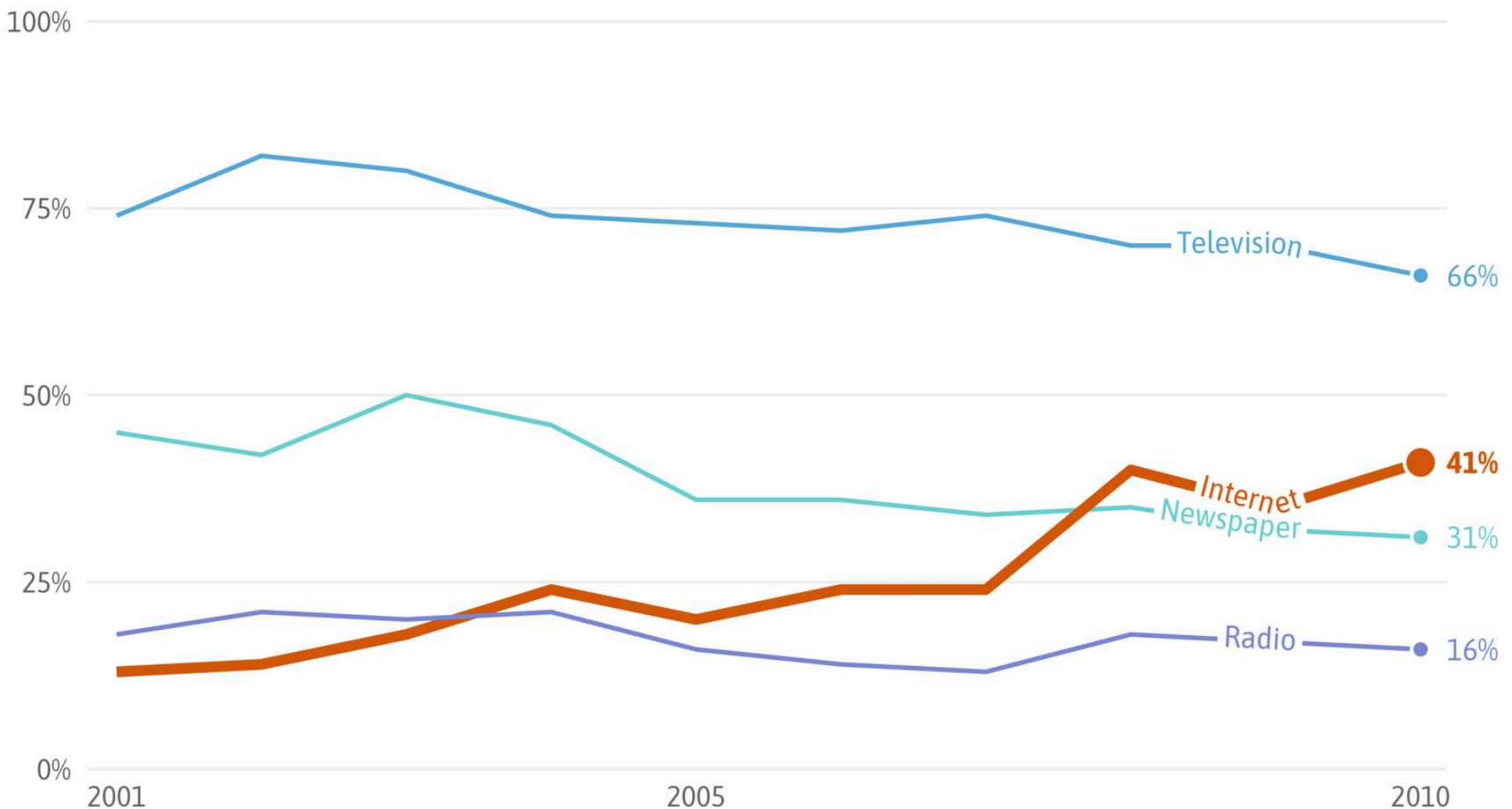
Responses to the question “where do you get most of your news about national and international issues?”



Data source: Pew Research Center
Figures sum to more than 100% because respondents could volunteer up to two main sources.

An increasing proportion cite the internet as their primary news source.

Responses to the question “where do you get most of your news about national and international issues?”

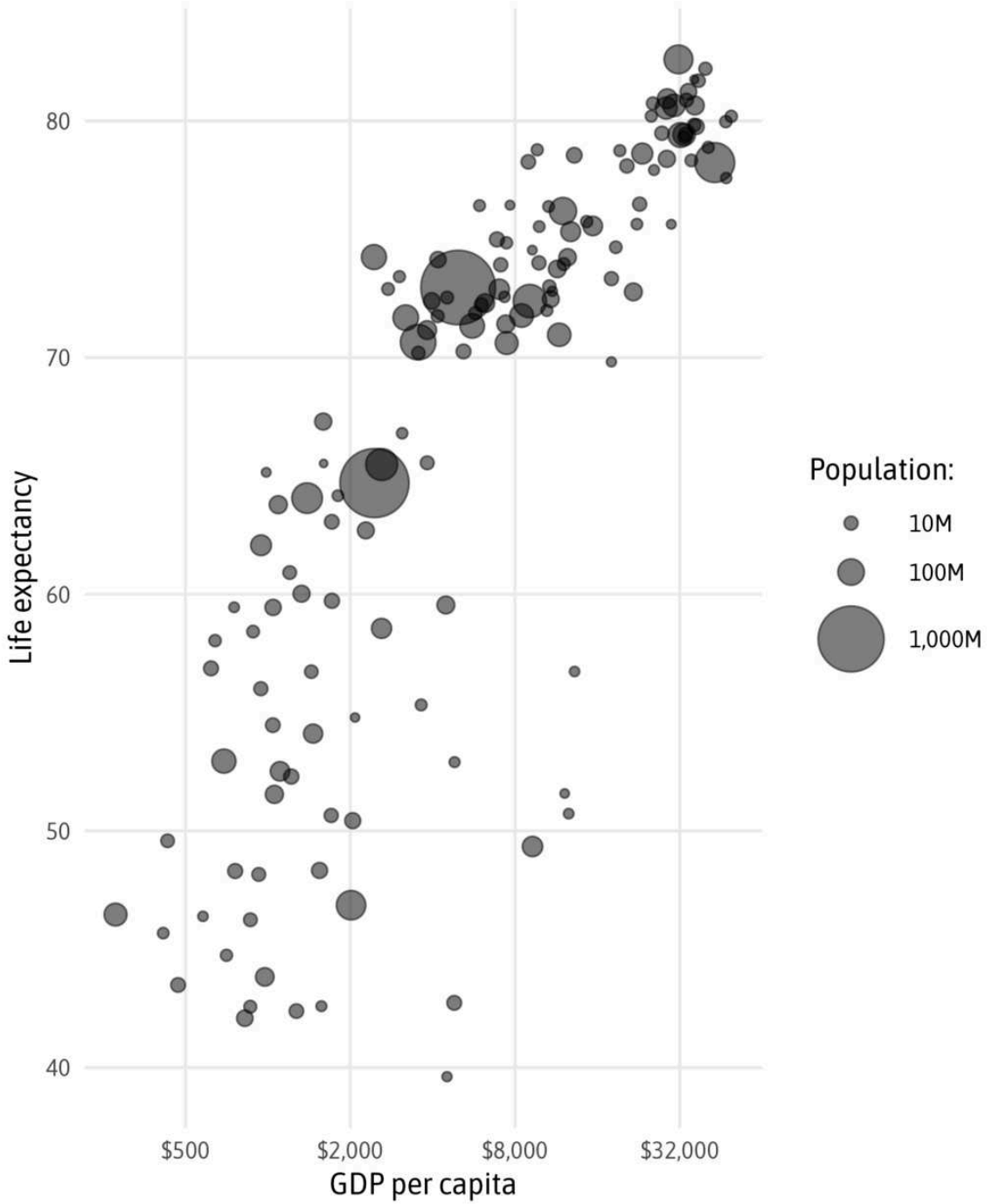


Data source: Pew Research Center
Figures sum to more than 100% because respondents could volunteer up to two main sources.

Exercise 2

- **Create a function that plots the famous Gapminder chart, highlighting one of the continents.**
  - Extend the code in `exercise-gapminder.qmd` to annotate a continent your choice of with {ggforce}.
  - Turn the code into a function with the utility to annotate any continent.
  - Optional: Create a second function to highlight a country.

```r
# install.packages("gapminder")

gm2007 <- filter(gapminder::gapminder, year == 2007)

ggplot(gm2007, aes(x = gdpPercap, y = lifeExp)) +
  geom_point(
    aes(size = pop), alpha = .5
  ) +
  scale_x_log10(
    breaks = c(500, 2000, 8000, 32000),
    labels = scales::label_dollar(accuracy = 1)
  ) +
  scale_size(
    range = c(1, 12), name = "Population:",
    breaks = c(10, 100, 1000)*1000000,
    labels = scales::label_comma(scale = 1 / 10^6, suffix = "M")
  ) +
  labs(x = "GDP per capita", y = "Life expectancy") +
  theme_minimal(base_family = "Asap SemiCondensed") +
  theme(panel.grid.minor = element_blank())
```


80
70
60
50
40
Life expectancy
$500
$2,000
$8,000
$32,000
GDP per capita
Population:
10M
100M
1,000M

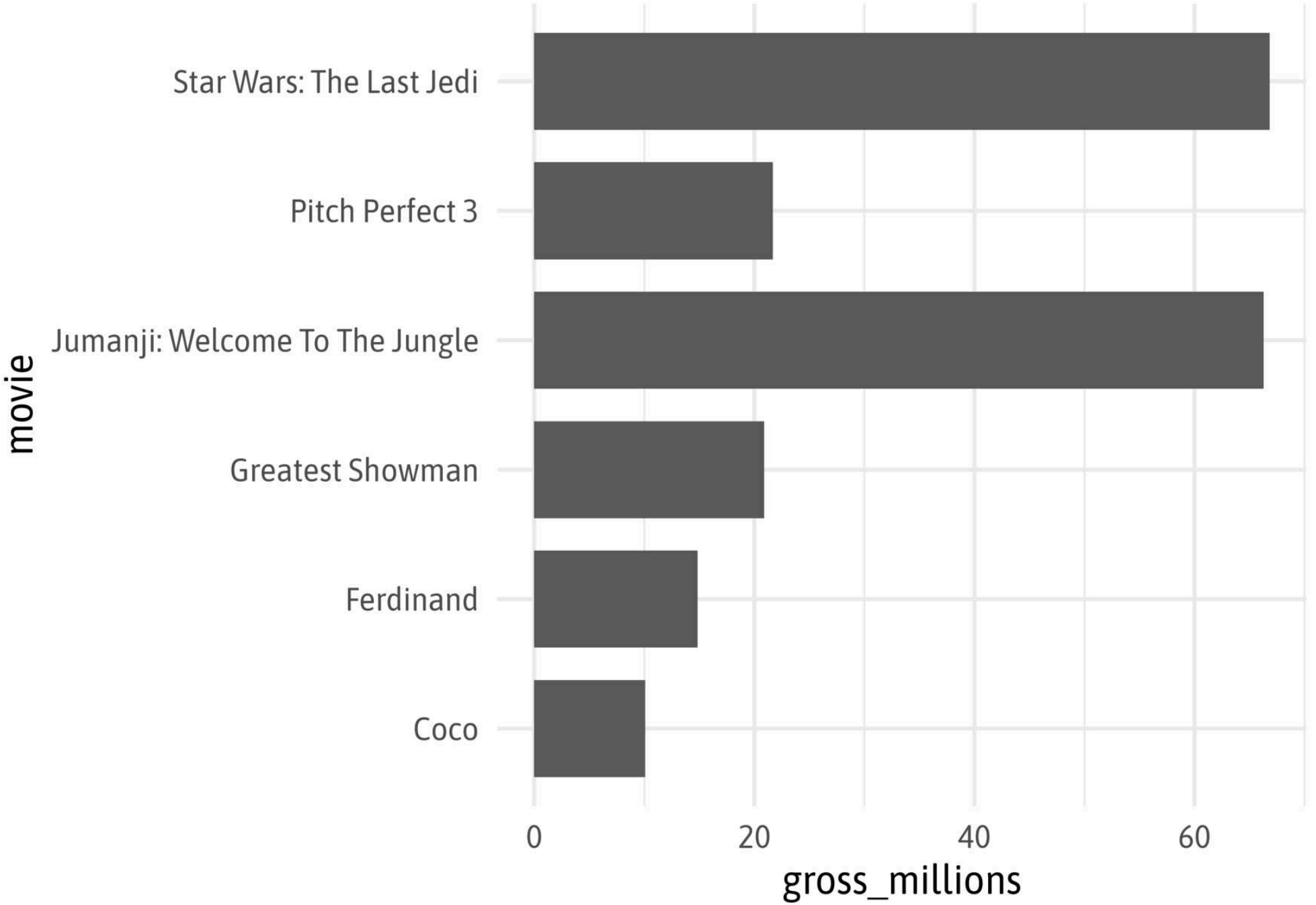
Solution Exercise 1

Cédric Scherer Data Visualization & Information Design

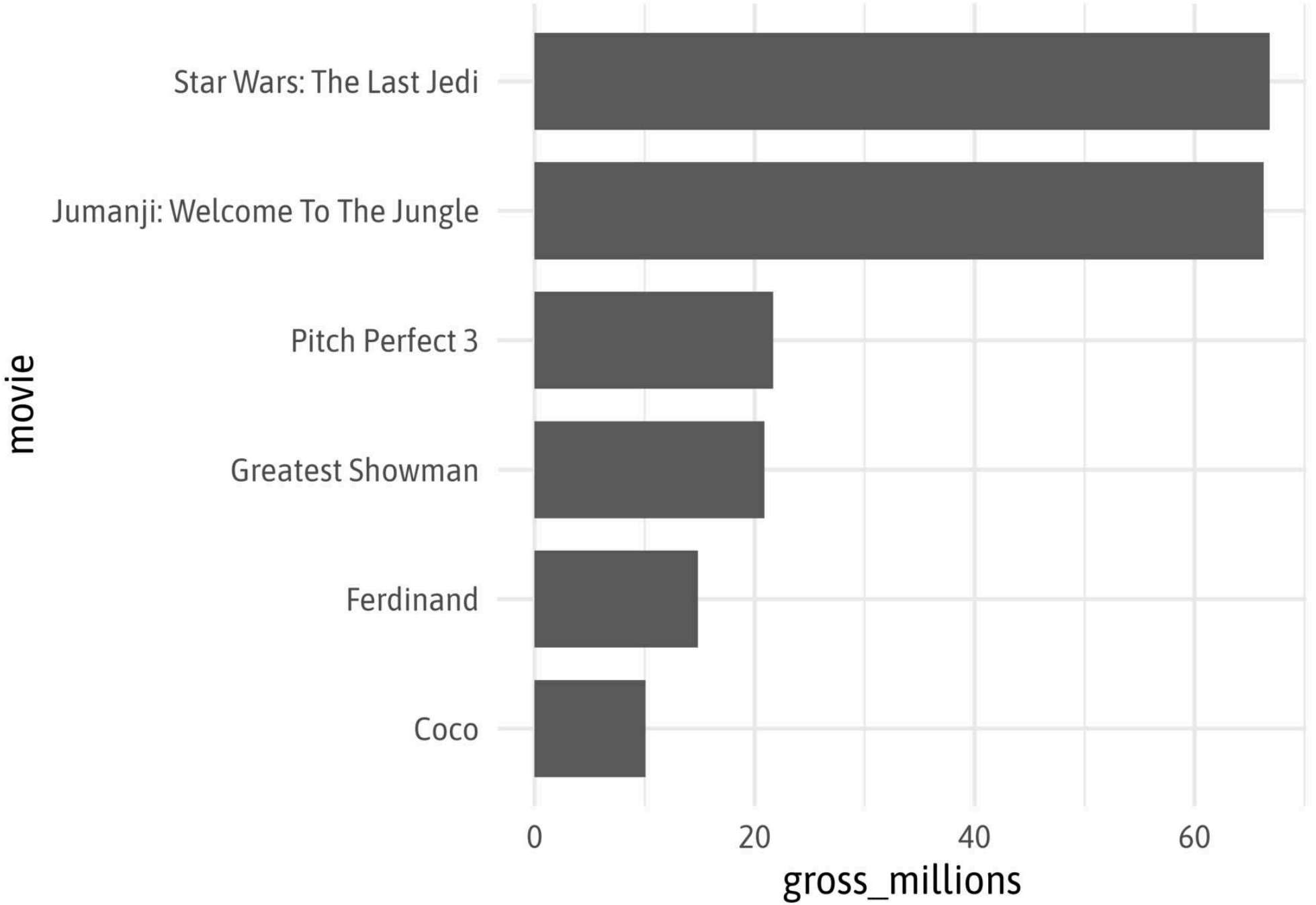
Exercise 1: Movies

```r
movies <- data.frame(
  movie = c("Star Wars: The Last Jedi", "Jumanji: Welcome To The Jungle",
            "Pitch Perfect 3", "Greatest Showman", "Ferdinand", "Coco"),
  gross_millions = c(66.814, 66.273, 21.676, 20.907, 14.852, 10.083)
)
```

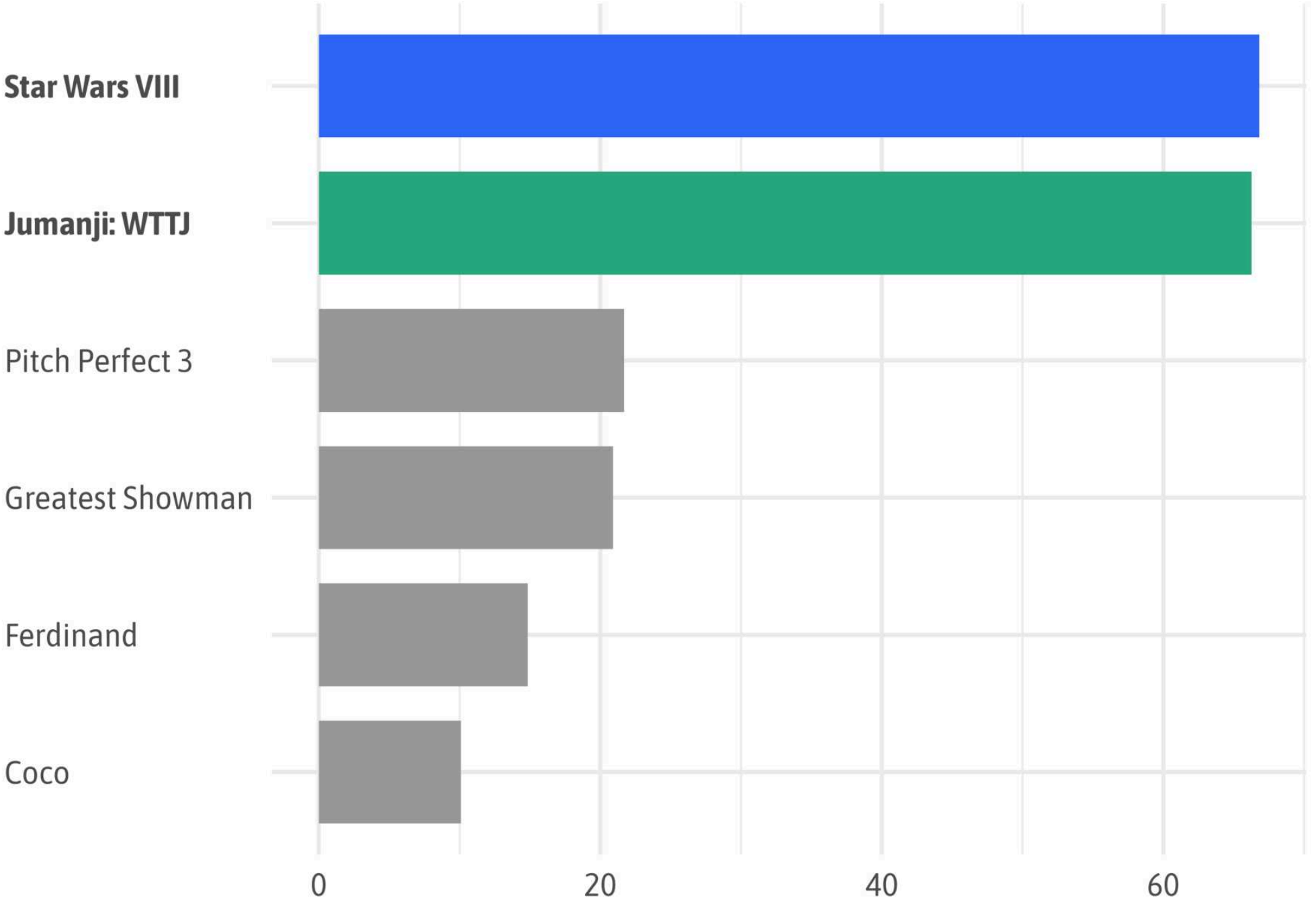
```r
ggplot(movies, aes(x = gross_millions, y = movie, gross_millions)) +
  geom_col(width = .75) +
  theme_minimal(base_size = 22, base_family = "Asap SemiCondensed")
```


Star Wars: The Last Jedi
Pitch Perfect 3
Jumanji: Welcome To The Jungle
Greatest Showman
Ferdinand
Coco
movie
0
20
40
60
gross_millions

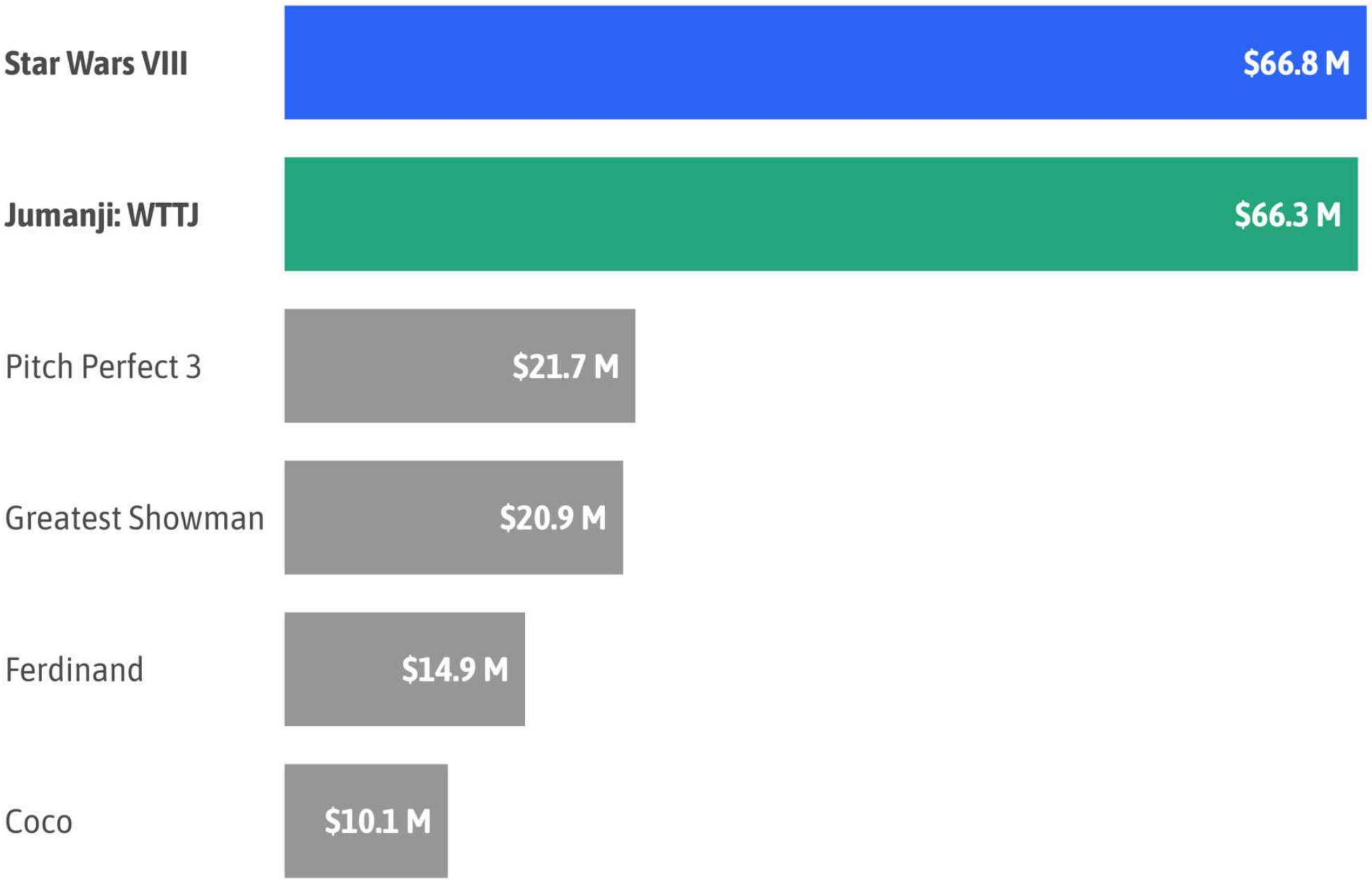
```r
movies |>
  mutate(movie = forcats::fct_reorder(movie, gross_millions)) |>
  ggplot(aes(x = gross_millions, y = movie)) +
  geom_col(width = .75) +
  theme_minimal(base_size = 22, base_family = "Asap SemiCondensed")
```


Star Wars: The Last Jedi
Jumanji: Welcome To The Jungle
Pitch Perfect 3
Greatest Showman
Ferdinand
Coco
movie
0
20
40
60
gross_millions

```r
movies |>
  mutate(movie = forcats::fct_reorder(movie, gross_millions)) |>
  ggplot(aes(x = gross_millions, y = movie)) +
  geom_col(aes(fill = movie), width = .75) +
  scale_y_discrete(labels = c("Coco", "Ferdinand", "Greatest Showman",
                              "Pitch Perfect 3", "Jumanji: WTTJ", "Star Wars VIII")) +
  scale_fill_manual(values = c(rep("grey60", 4), "#28A87D", "#2E67F8"), guide = "none"
  labs(y = NULL, x = NULL) +
  theme_minimal(base_size = 22, base_family = "Asap SemiCondensed") +
  theme(
    axis.text.y = element_text(hjust = 0, face = c(rep("plain", 4), "bold", "bold"))
  )
```

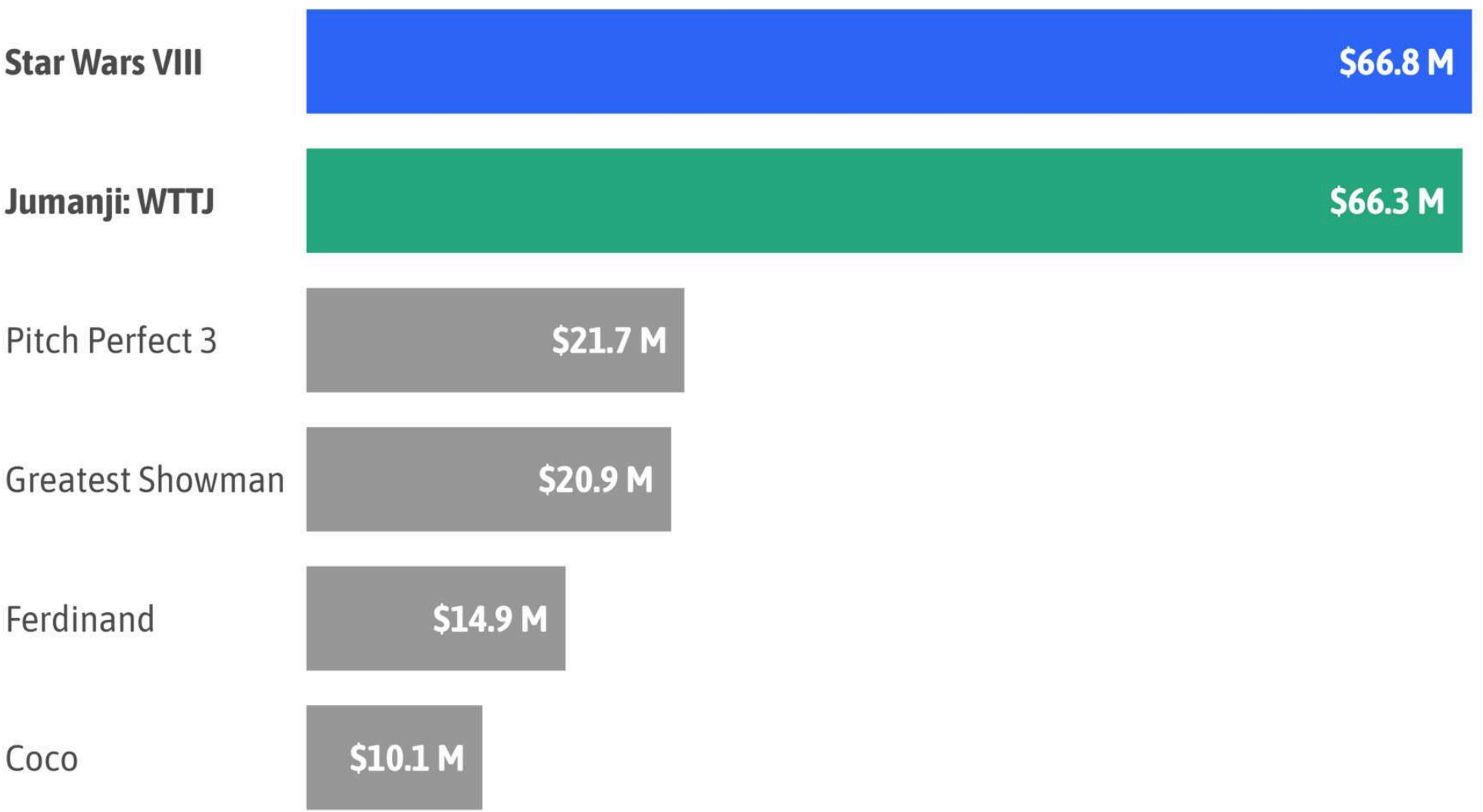

Star Wars VIII
Jumanji: WTTJ
Pitch Perfect 3
Greatest Showman
Ferdinand
Coco
0
20
40
60

```r
movies |>
  mutate(movie = forcats::fct_reorder(movie, gross_millions)) |>
  ggplot(aes(x = gross_millions, y = movie)) +
  geom_col(aes(fill = movie), width = .75) +
  geom_text(
    aes(label = paste0("$", sprintf("%2.1f", gross_millions), " M")),
    hjust = 1.15, color = "white", family = "Asap SemiCondensed", fontface = "bold", s
  ) +
  scale_x_continuous(expand = c(0, 0), guide = "none") +
  scale_y_discrete(labels = c("Coco", "Ferdinand", "Greatest Showman",
                              "Pitch Perfect 3", "Jumanji: WTTJ", "Star Wars VIII")) +
  scale_fill_manual(values = c(rep("grey60", 4), "#28A87D", "#2E67F8"), guide = "none"
  labs(y = NULL, x = NULL) +
  theme_minimal(base_size = 22, base_family = "Asap SemiCondensed") +
  theme(
    axis.text.y = element_text(hjust = 0, face = c(rep("plain", 4), rep("bold", 2))),
    panel.grid = element_blank()
  )
```


Star Wars VIII
$66.8 M
Jumanji: WTTJ
$66.3 M
Pitch Perfect 3
$21.7 M
Greatest Showman
$20.9 M
Ferdinand
$14.9 M
Coco
$10.1 M

```r
movies |>
  mutate(movie = forcats::fct_reorder(movie, gross_millions)) |>
  ggplot(aes(x = gross_millions, y = movie)) +
  geom_col(aes(fill = movie), width = .75) +
  geom_text(
    aes(label = paste0("$", sprintf("%2.1f", gross_millions), " M")),
    hjust = 1.15, color = "white", family = "Asap SemiCondensed", fontface = "bold", s
  ) +
  scale_x_continuous(expand = c(0, 0), guide = "none") +
  scale_y_discrete(labels = c("Coco", "Ferdinand", "Greatest Showman",
                              "Pitch Perfect 3", "Jumanji: WTTJ", "Star Wars VIII")) +
  scale_fill_manual(values = c(rep("grey60", 4), "#28A87D", "#2E67F8"), guide = "none"
  labs(y = NULL, x = NULL,
       title = '<b style="color:#2E67F8;">“Stars Wars VIII - The Last Jedi”</b> just b
  theme_minimal(base_size = 22, base_family = "Asap SemiCondensed") +
  theme(
    axis.text.y = element_text(hjust = 0, face = c(rep("plain", 4), rep("bold", 2))),
    panel.grid = element_blank(),
    plot.title = ggtext::element_markdown(
      size = rel(1.1), face = "bold", lineheight = 1.2, margin = margin(b = 20)
    ),
    plot.title.position = "plot"
  )
```

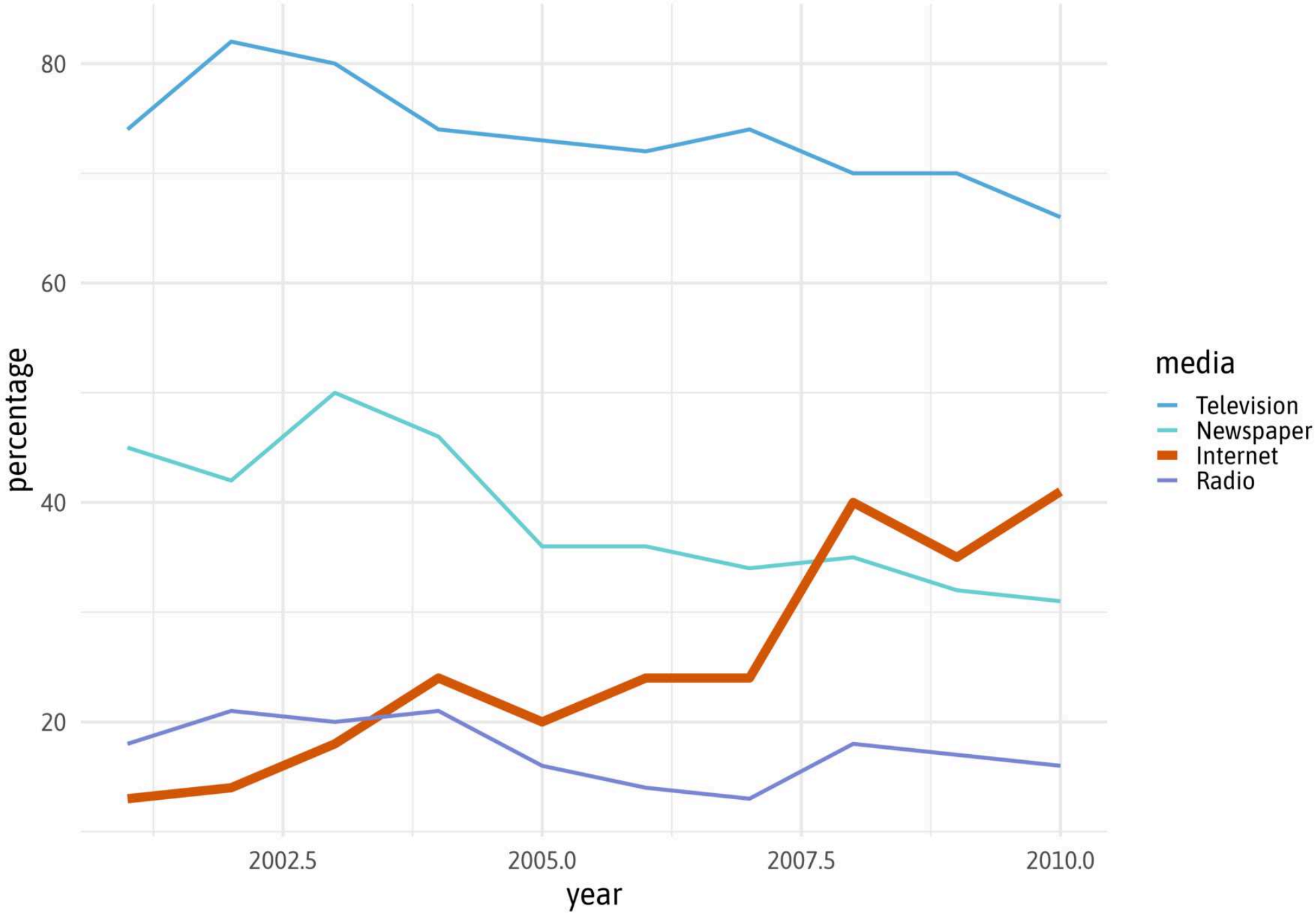
“Stars Wars VIII – The Last Jedi” just barely topped the box office this weel facing stiff competition from “Jumanji: Welcome to the Jungle”.



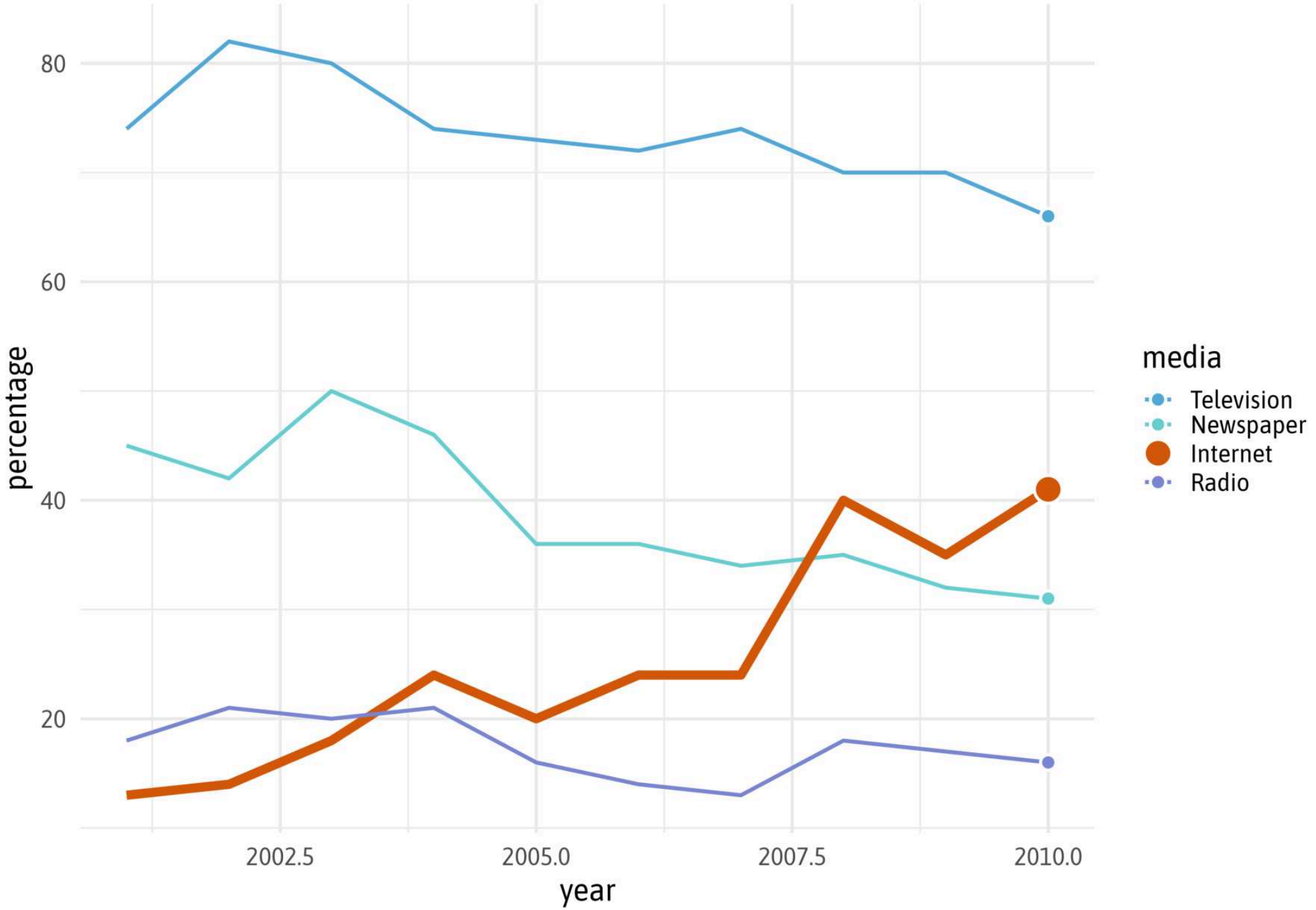
Exercise 1: News

```r
data <-
  tibble::tribble(
    ~year, ~Television, ~Newspaper, ~Internet, ~Radio,
    2001, 74, 45, 13, 18,
    2002, 82, 42, 14, 21,
    2003, 80, 50, 18, 20,
    2004, 74, 46, 24, 21,
    2005, 73, 36, 20, 16,
    2006, 72, 36, 24, 14,
    2007, 74, 34, 24, 13,
    2008, 70, 35, 40, 18,
    2009, 70, 32, 35, 17,
    2010, 66, 31, 41, 16
  ) |>
  tidyr::pivot_longer(
    cols = -c(year), names_to = "media", values_to = "percentage"
  ) |>
  mutate(media = forcats::fct_inorder(media))

title <- "An increasing proportion cite the <b style='color:#d4580a;'>internet</b> as
subtitle <- 'Responses to the question “where do you get most of your news about natio
caption <- 'Data source: Pew Research Center\nFigures sum to more than 100% because re
```

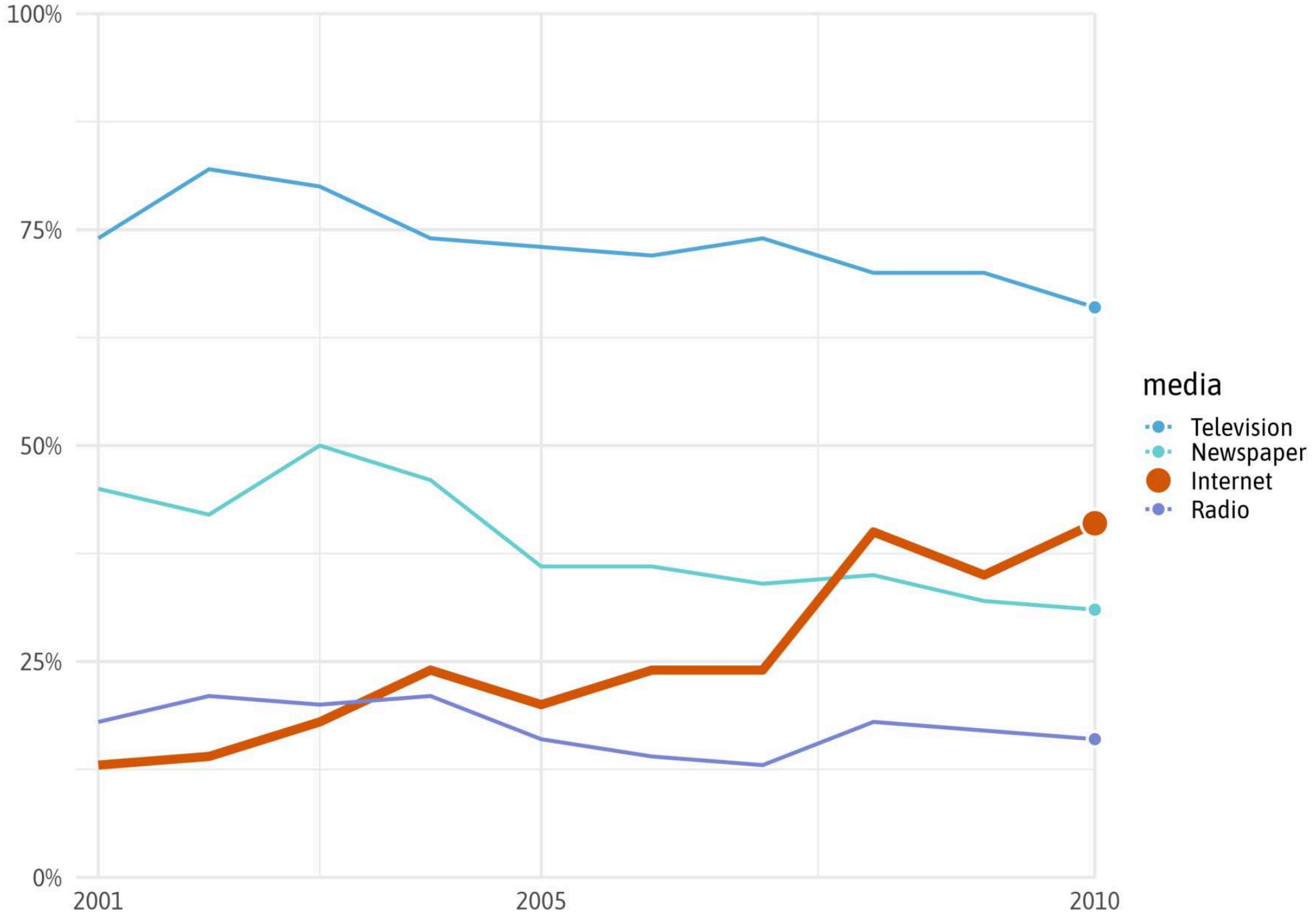
```r
ggplot(news, aes(x = year, y = percentage, group = media)) +
  geom_line(aes(color = media, linewidth = media)) +
  scale_color_manual(values = c("#54A9D9", "#6ACFD2", "#d4580a", "#7b87d2")) +
  scale_linewidth_manual(values = c(1.3, 1.3, 3, 1.3)) +
  theme_minimal(base_size = 22, base_family = "Asap SemiCondensed")
```


80
60
40
20
percentage
2002.5
2005.0
2007.5
2010.0
year
media
Television
Newspaper
Internet
Radio

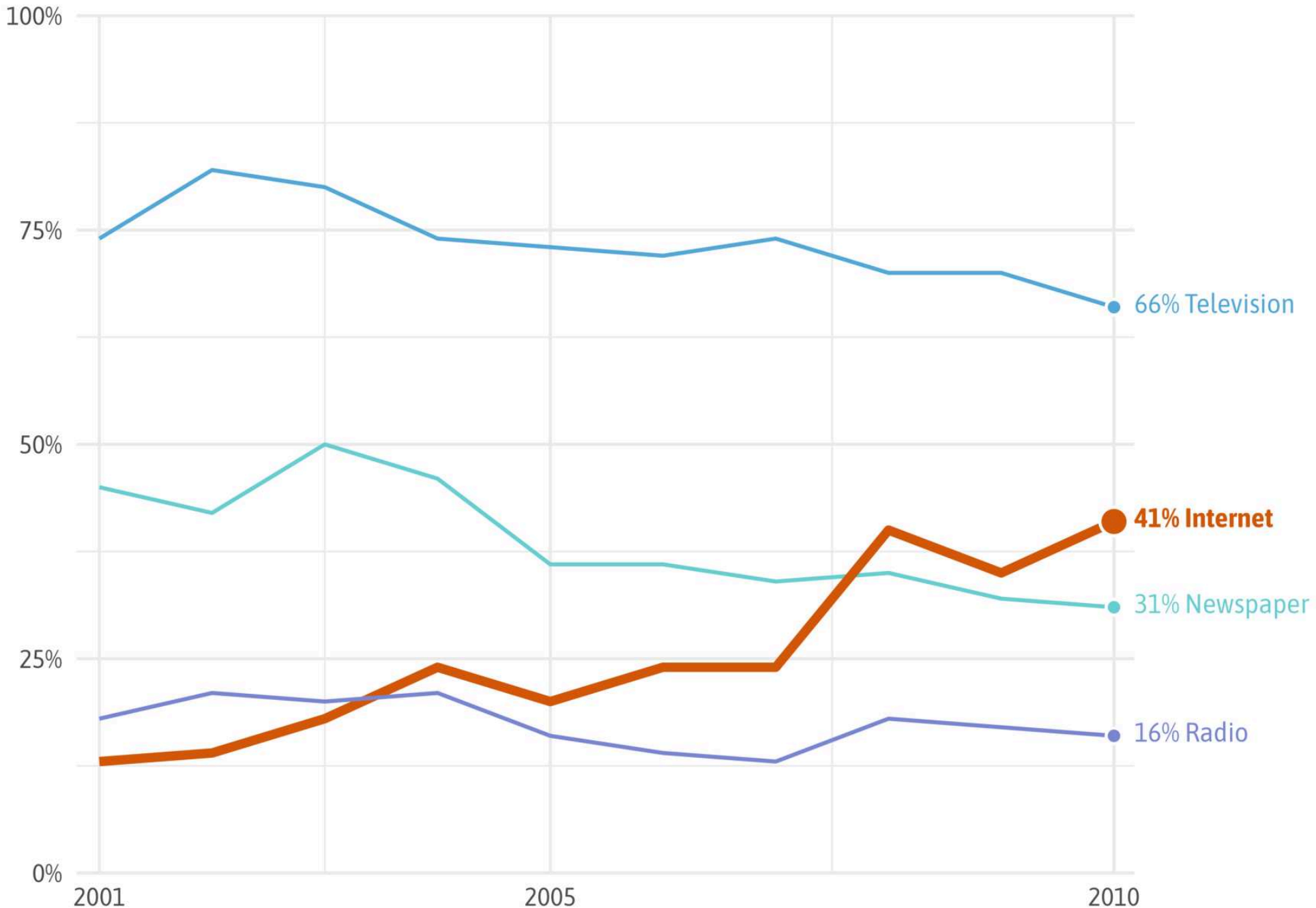
```r
ggplot(news, aes(x = year, y = percentage, group = media)) +
  geom_line(aes(color = media, linewidth = media)) +
  geom_point(
    data = filter(news, year == 2010),
    aes(fill = media, size = media),
    shape = 21, stroke = 1.5, color = "white"
  ) +
  scale_color_manual(values = c("#54A9D9", "#6ACFD2", "#d4580a", "#7b87d2")) +
  scale_fill_manual(values = c("#54A9D9", "#6ACFD2", "#d4580a", "#7b87d2")) +
  scale_size_manual(values = c(4, 4, 8, 4)) +
  scale_linewidth_manual(values = c(1.3, 1.3, 3, 1.3)) +
  theme_minimal(base_size = 22, base_family = "Asap SemiCondensed")
```


80
60
40
20
percentage
2002.5
2005.0
2007.5
2010.0
year
media
Television
Newspaper
Internet
Radio

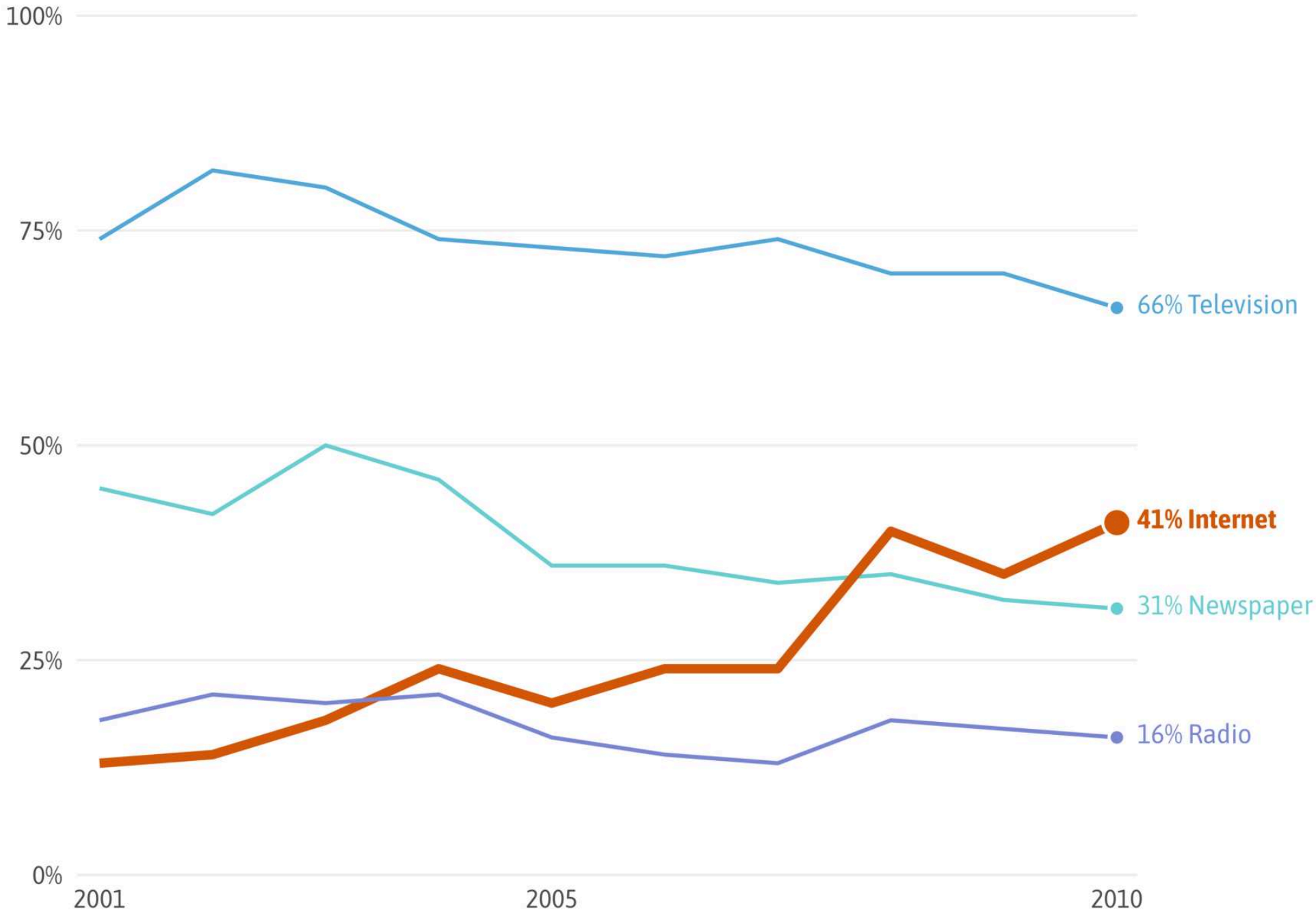
```r
ggplot(news, aes(x = year, y = percentage, group = media)) +
  geom_line(aes(color = media, linewidth = media)) +
  geom_point(
    data = filter(news, year == 2010), aes(fill = media, size = media),
    shape = 21, stroke = 1.5, color = "white"
  ) +
  coord_cartesian(clip = "off") +
  scale_x_continuous(
    breaks = c(2001, 2005, 2010), expand = expansion(add = c(.2, 0)), name = NULL
  ) +
  scale_y_continuous(
    breaks = 0:4*25, labels = scales::label_percent(scale = 1),
    limits = c(0, 100), expand = c(0, 0), name = NULL
  ) +
  scale_color_manual(values = c("#54A9D9", "#6ACFD2", "#d4580a", "#7b87d2")) +
  scale_fill_manual(values = c("#54A9D9", "#6ACFD2", "#d4580a", "#7b87d2")) +
  scale_size_manual(values = c(4, 4, 8, 4)) +
  scale_linewidth_manual(values = c(1.3, 1.3, 3, 1.3)) +
  theme_minimal(base_size = 22, base_family = "Asap SemiCondensed")
```


100%
75%
50%
25%
0%
2001
2005
2010
media
Television
Newspaper
Internet
Radio

```r
ggplot(news, aes(x = year, y = percentage, group = media)) +
  geom_line(aes(color = media, linewidth = media)) +
  geom_point(
    data = filter(news, year == 2010), aes(fill = media, size = media),
    shape = 21, stroke = 1.5, color = "white"
  ) +
  geom_text(
    data = filter(news, year == 2010),
    aes(label = paste0(percentage, "% ", media), color = media),
    hjust = 0, nudge_x = .18, family = "Asap SemiCondensed", size = 6.5,
    fontface = c("plain", "plain", "bold", "plain"), vjust = .3
  ) +
  coord_cartesian(clip = "off") +
  scale_x_continuous(
    breaks = c(2001, 2005, 2010), expand = expansion(add = c(.2, 0)), name = NULL
  ) +
  scale_y_continuous(
    breaks = 0:4*25, labels = scales::label_percent(scale = 1),
    limits = c(0, 100), expand = c(0, 0), name = NULL
  ) +
  scale_color_manual(values = c("#54A9D9", "#6ACFD2", "#d4580a", "#7b87d2")) +
  scale_fill_manual(values = c("#54A9D9", "#6ACFD2", "#d4580a", "#7b87d2")) +
  scale_size_manual(values = c(4, 4, 8, 4)) +
  scale_linewidth_manual(values = c(1.3, 1.3, 3, 1.3)) +
```


100%
75%
50%
25%
0%
2001
2005
2010
66% Television
41% Internet
31% Newspaper
16% Radio

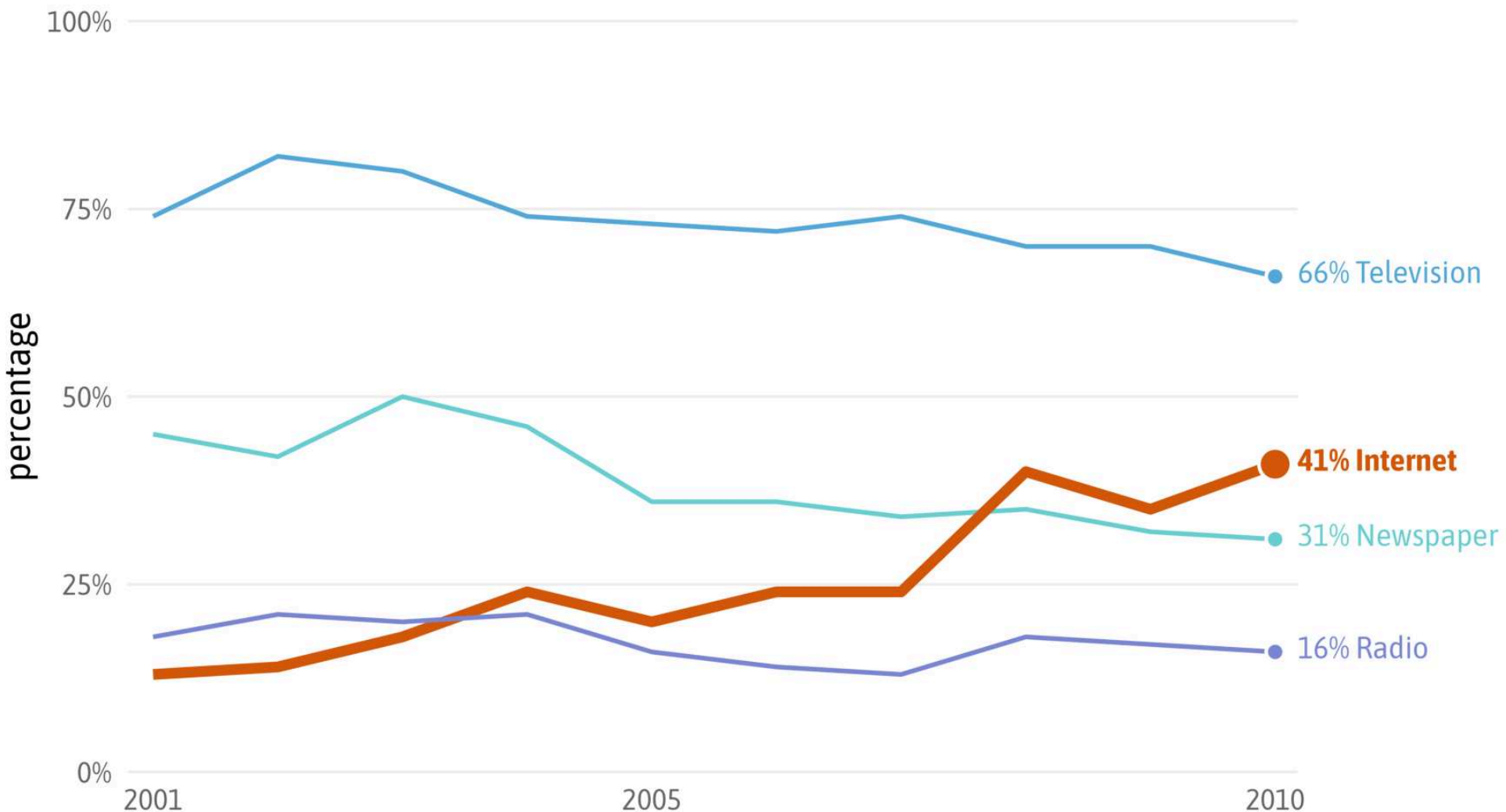
```r
ggplot(news, aes(x = year, y = percentage, group = media)) +
  geom_line(aes(color = media, linewidth = media)) +
  geom_point(
    data = filter(news, year == 2010), aes(fill = media, size = media),
    shape = 21, stroke = 1.5, color = "white"
  ) +
  geom_text(
    data = filter(news, year == 2010),
    aes(label = paste0(percentage, "% ", media), color = media),
    hjust = 0, nudge_x = .18, family = "Asap SemiCondensed", size = 6.5,
    fontface = c("plain", "plain", "bold", "plain"), vjust = .3
  ) +
  coord_cartesian(clip = "off") +
  scale_x_continuous(
    breaks = c(2001, 2005, 2010), expand = expansion(add = c(.2, 0)), name = NULL
  ) +
  scale_y_continuous(
    breaks = 0:4*25, labels = scales::label_percent(scale = 1),
    limits = c(0, 100), expand = c(0, 0), name = NULL
  ) +
  scale_color_manual(values = c("#54A9D9", "#6ACFD2", "#d4580a", "#7b87d2")) +
  scale_fill_manual(values = c("#54A9D9", "#6ACFD2", "#d4580a", "#7b87d2")) +
  scale_size_manual(values = c(4, 4, 8, 4)) +
  scale_linewidth_manual(values = c(1.3, 1.3, 3, 1.3)) +
```


100%
75%
50%
25%
0%
2001
2005
2010
66% Television
41% Internet
31% Newspaper
16% Radio

```r
ggplot(news, aes(x = year, y = percentage, group = media)) +
  geom_line(aes(color = media, linewidth = media)) +
  geom_point(
    data = filter(news, year == 2010), aes(fill = media, size = media),
    shape = 21, stroke = 1.5, color = "white"
  ) +
  geom_text(
    data = filter(news, year == 2010),
    aes(label = paste0(percentage, "% ", media), color = media),
    hjust = 0, nudge_x = .18, family = "Asap SemiCondensed", size = 6.5,
    fontface = c("plain", "plain", "bold", "plain"), vjust = .3
  ) +
  coord_cartesian(clip = "off") +
  scale_x_continuous(
    breaks = c(2001, 2005, 2010), expand = expansion(add = c(.2, 0)), name = NULL
  ) +
  scale_y_continuous(
    breaks = 0:4*25, labels = scales::label_percent(scale = 1),
    limits = c(0, 100), expand = c(0, 0)
  ) +
  scale_color_manual(values = c("#54A9D9", "#6ACFD2", "#d4580a", "#7b87d2")) +
  scale_fill_manual(values = c("#54A9D9", "#6ACFD2", "#d4580a", "#7b87d2")) +
  scale_size_manual(values = c(4, 4, 8, 4)) +
  scale_linewidth_manual(values = c(1.3, 1.3, 3, 1.3)) +
```

An increasing proportion cite the internet as their primary news source.

Responses to the question "where do you get most of your news about national and international issues?"



Data source: Pew Research Center
Figures sum to more than 100% because respondents could volunteer up to two main sources.

Solution Exercise 2

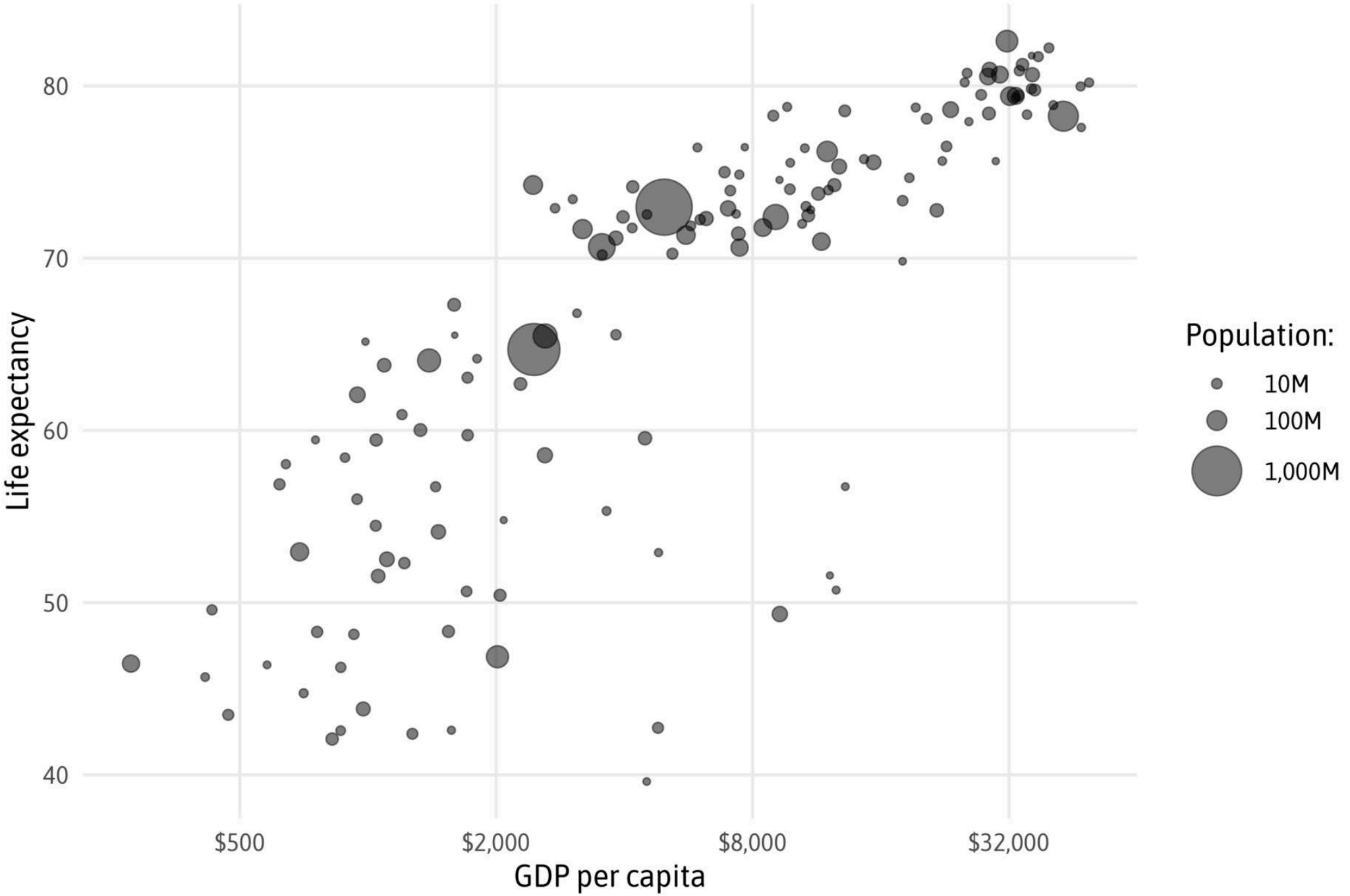
Cédric Scherer Data Visualization & Information Design

Exercise 2

- **Create a function that plots the famous Gapminder chart, highlighting one of the continents.**
  - Extend the code in `exercise-gapminder.qmd` to annotate a continent your choice of with {ggforce}.
  - Turn the code into a function with the utility to annotate any continent.
  - Optional: Create a second function to highlight a country.

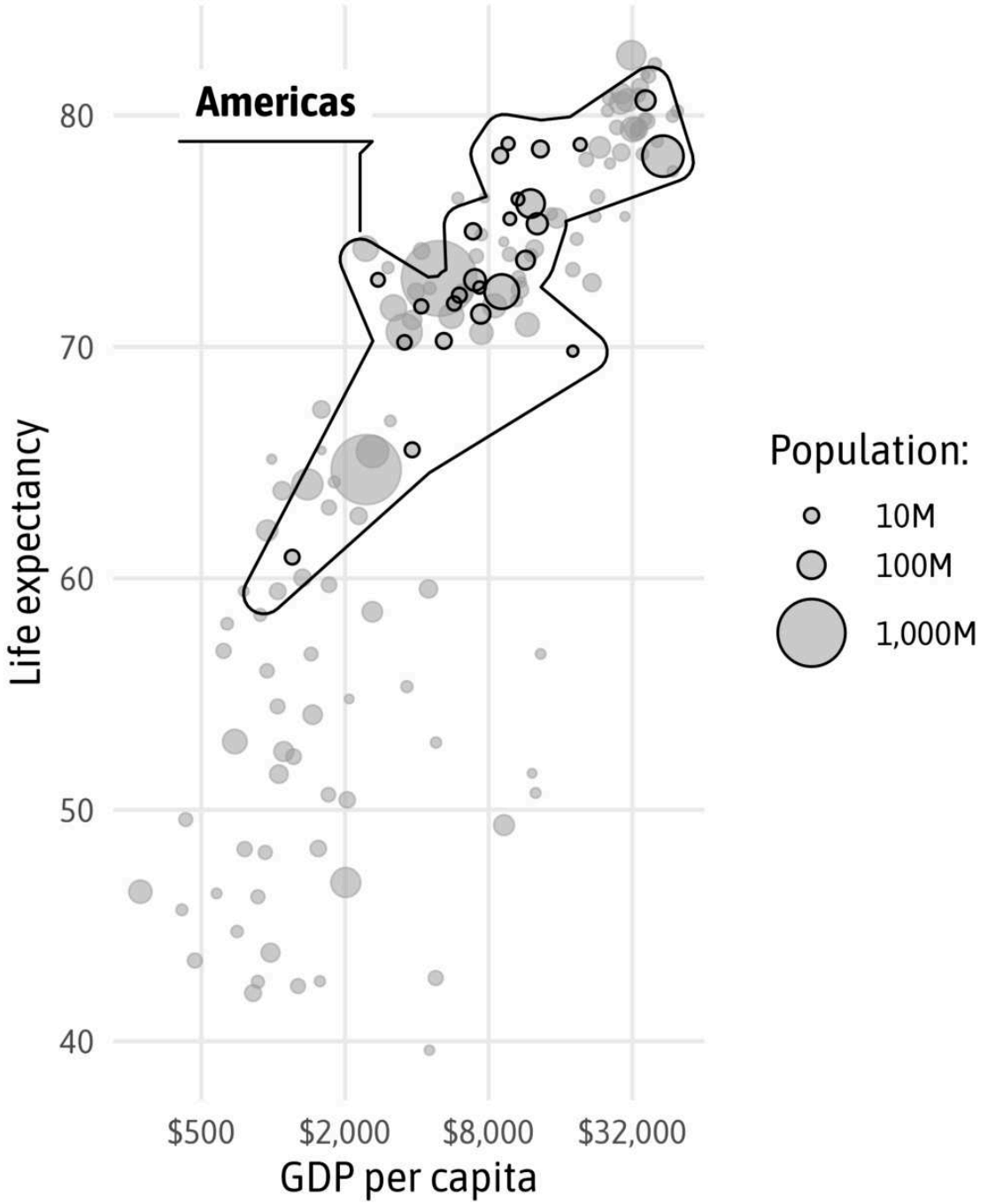
Gapminder Bubble Chart

```r
# install.packages("gapminder")

library(gapminder)
library(dplyr)
library(ggplot2)

gm2007 <- filter(gapminder, year == 2007)

ggplot(gm2007, aes(x = gdpPercap, y = lifeExp)) +
  geom_point(
    aes(size = pop), alpha = .5
  ) +
  scale_x_log10(
    breaks = c(500, 2000, 8000, 32000),
    labels = scales::label_dollar(accuracy = 1)
  ) +
  scale_size(
    range = c(1, 12), name = "Population:",
    breaks = c(10, 100, 1000)*1000000,
    labels = scales::label_comma(scale = 1 / 10^6, suffix = "M")
  ) +
  labs(x = "GDP per capita", y = "Life expectancy") +
  theme_minimal(base_family = "Asap SemiCondensed", base_size = 15) +
  theme(panel.grid.minor = element_blank())
```


80
70
60
50
40
Life expectancy
$500
$2,000
$8,000
$32,000
GDP per capita
Population:
10M
100M
1,000M

Annotate Continents

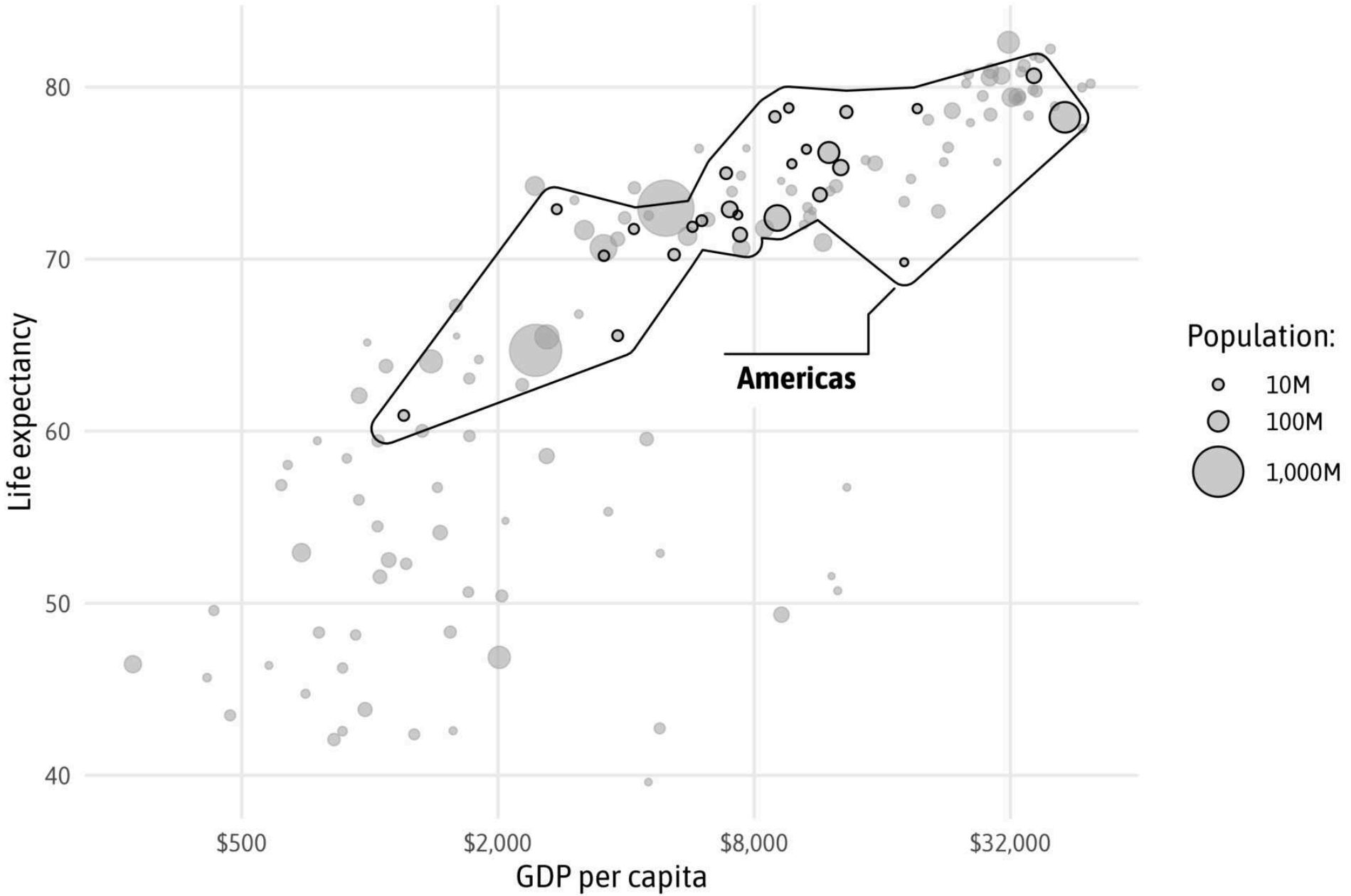
```
ggplot(gm2007, aes(x = gdpPercap, y = lifeExp)) +
  geom_point(
    aes(size = pop), alpha = .5, color = "grey60"
  ) +
  geom_point(
    data = filter(gm2007, continent == "Americas"),
    aes(size = pop), shape = 1, stroke = .7
  ) +
  ggforce::geom_mark_hull(
    aes(label = continent, filter = continent == "Americas"),
    expand = unit(10, "pt"), con.cap = unit(1, "mm"),
    label.family = "Asap SemiCondensed", label.fontsize = 15
  ) +
  scale_x_log10(
    breaks = c(500, 2000, 8000, 32000),
    labels = scales::label_dollar(accuracy = 1)
  ) +
  scale_size(
    range = c(1, 12), name = "Population:",
    breaks = c(10, 100, 1000)*1000000,
    labels = scales::label_comma(scale = 1 / 10^6, suffix = "M")
  ) +
  labs(x = "GDP per capita", y = "Life expectancy") +
  theme_minimal(base_family = "Asap SemiCondensed", base_size = 15) +
```


Americas
80
70
60
50
40
Life expectancy
$500
$2,000
$8,000
$32,000
GDP per capita
Population:
10M
100M
1,000M

Function to Highlight a Continent

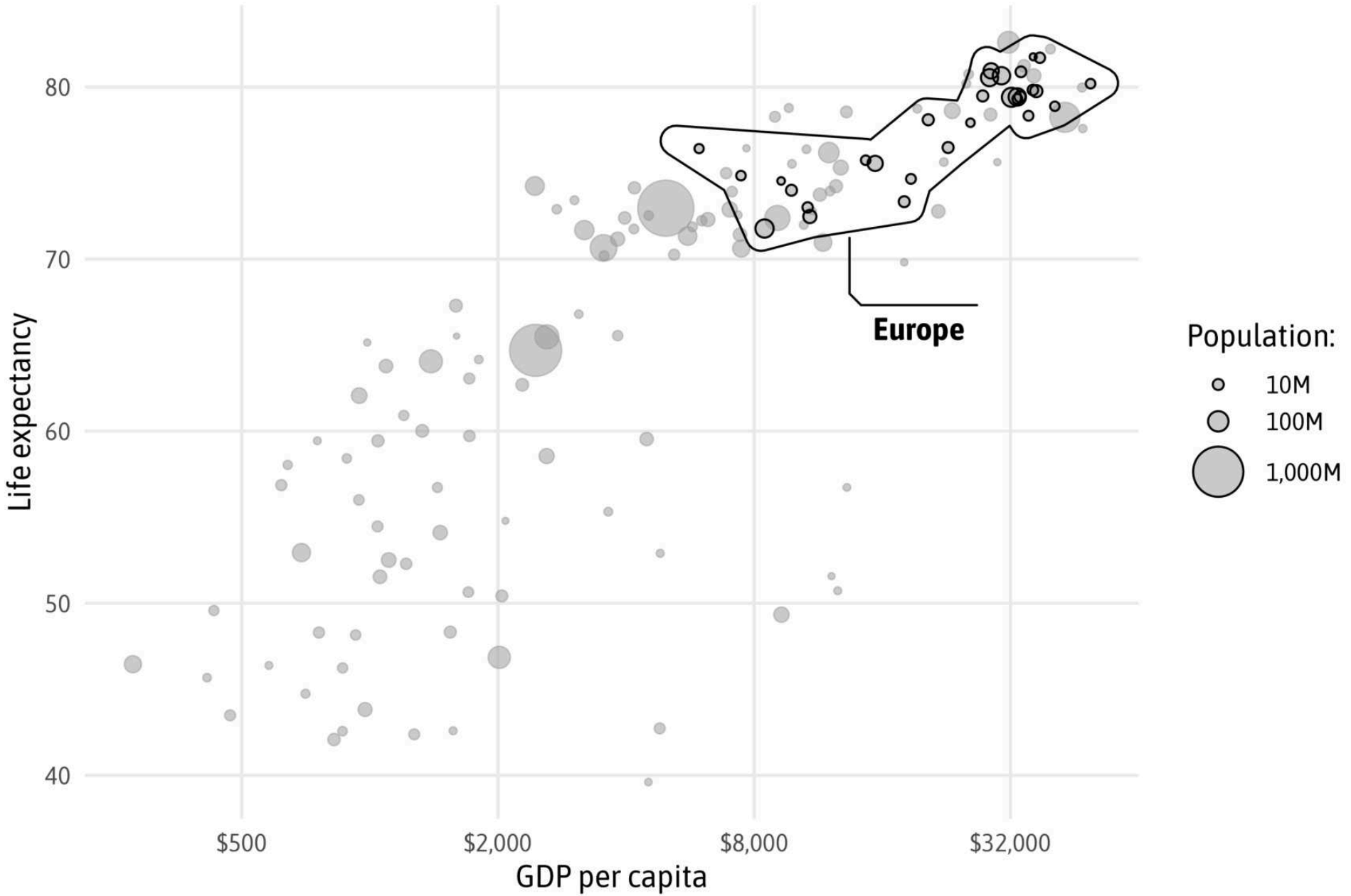
```
draw_gp_continent <- function(grp) {

  ggplot(gm2007, aes(x = gdpPercap, y = lifeExp)) +
    geom_point(
      aes(size = pop), alpha = .5, color = "grey60"
    ) +
    geom_point(
      data = filter(gm2007, continent == grp),
      aes(size = pop), shape = 1, stroke = .7
    ) +
    ggforce::geom_mark_hull(
      aes(label = continent, filter = continent == grp),
      expand = unit(10, "pt"), con.cap = unit(1, "mm"),
      label.family = "Asap SemiCondensed", label.fontsize = 15
    ) +
    scale_x_log10(
      breaks = c(500, 2000, 8000, 32000),
      labels = scales::label_dollar(accuracy = 1)
    ) +
    scale_size(
      range = c(1, 12), name = "Population:",
      breaks = c(10, 100, 1000)*1000000,
      labels = scales::label_comma(scale = 1 / 10^6, suffix = "M")
    ) +
```

```
draw_gp_continent("Americas")
```



```
draw_gp_continent("Africa")
```

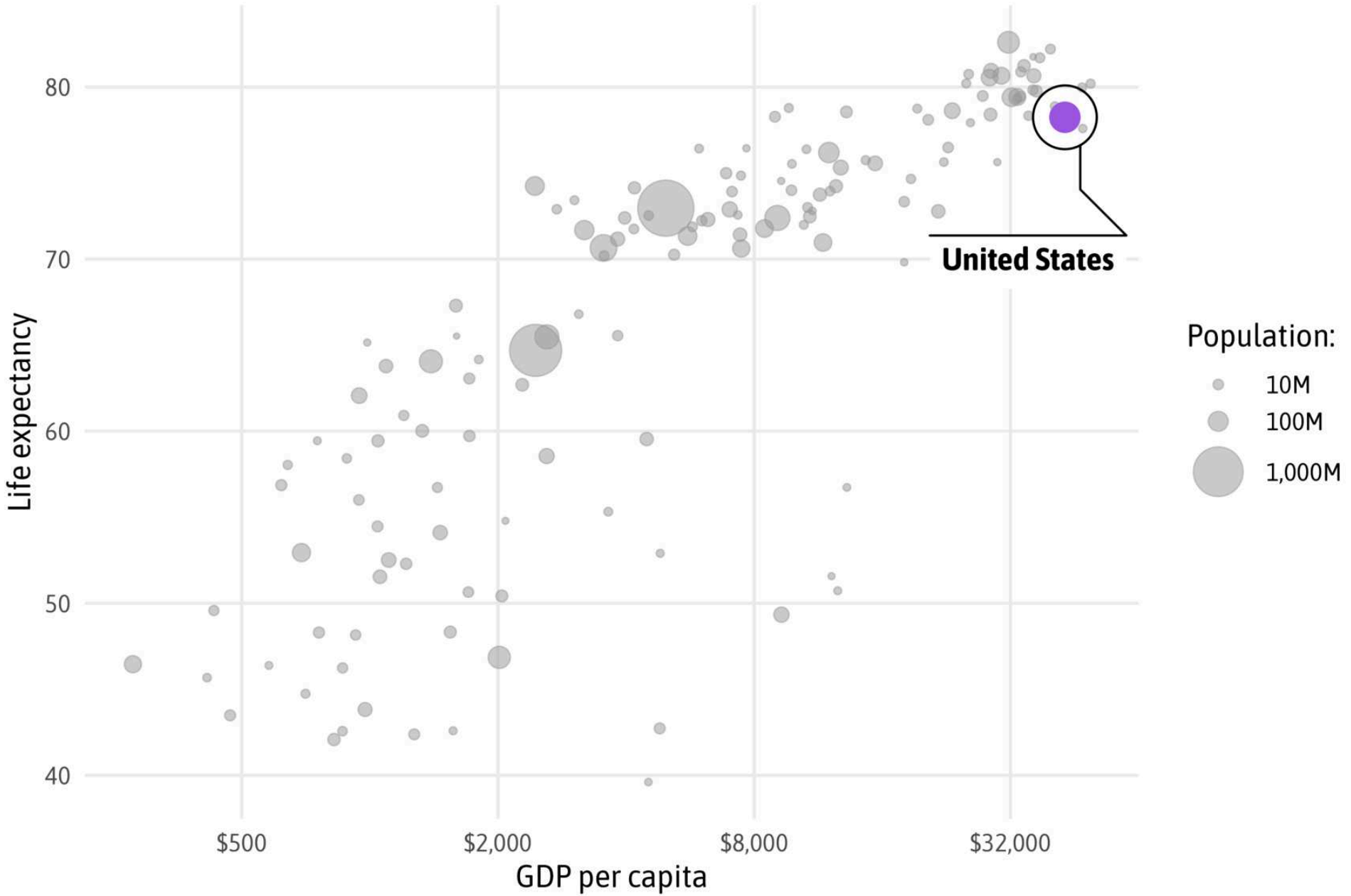
```
draw_gp_continent("Europe")
```



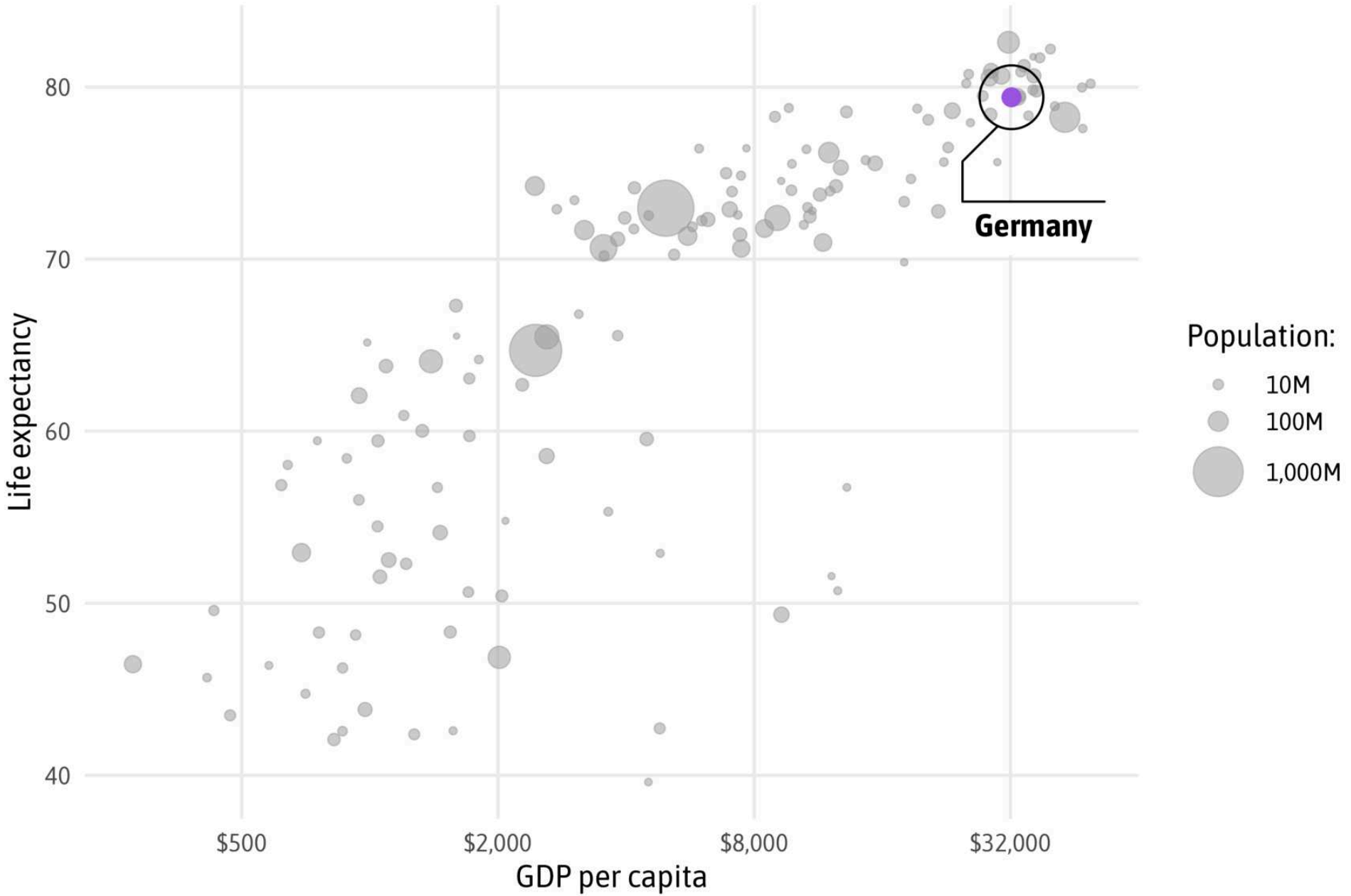
Function to Highlight a Country

```
draw_gp_country <- function(grp) {

  ggplot(gm2007, aes(x = gdpPercap, y = lifeExp)) +
    geom_point(
      aes(size = pop), alpha = .5, color = "grey60"
    ) +
    ggforce::geom_mark_circle(
      aes(label = country, filter = country == grp),
      expand = unit(15, "pt"), con.cap = unit(0, "mm"),
      # expand = unit(0, "pt"), con.cap = unit(0, "mm"),
      label.family = "Asap SemiCondensed", label.fontsize = 15
    ) +
    geom_point(
      data = filter(gm2007, country == grp),
      aes(size = pop), color = "#9C55E3", show.legend = FALSE
    ) +
    scale_x_log10(
      breaks = c(500, 2000, 8000, 32000),
      labels = scales::label_dollar(accuracy = 1)
    ) +
    scale_size(
      range = c(1, 12), name = "Population:",
      breaks = c(10, 100, 1000)*1000000,
      labels = scales::label_comma(scale = 1 / 1e06, suffix = "M")
```

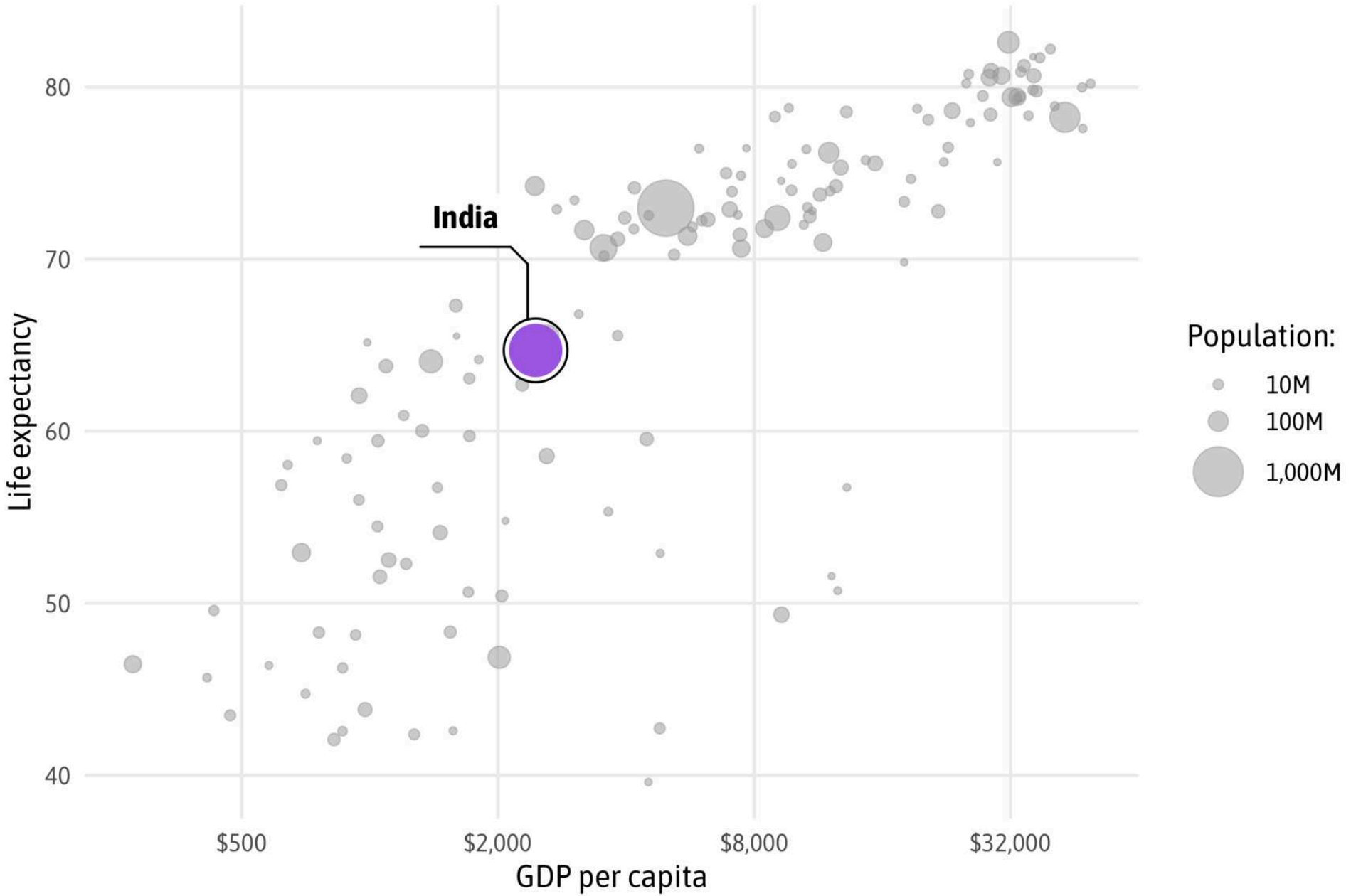
```
draw_gp_country("United States")
```



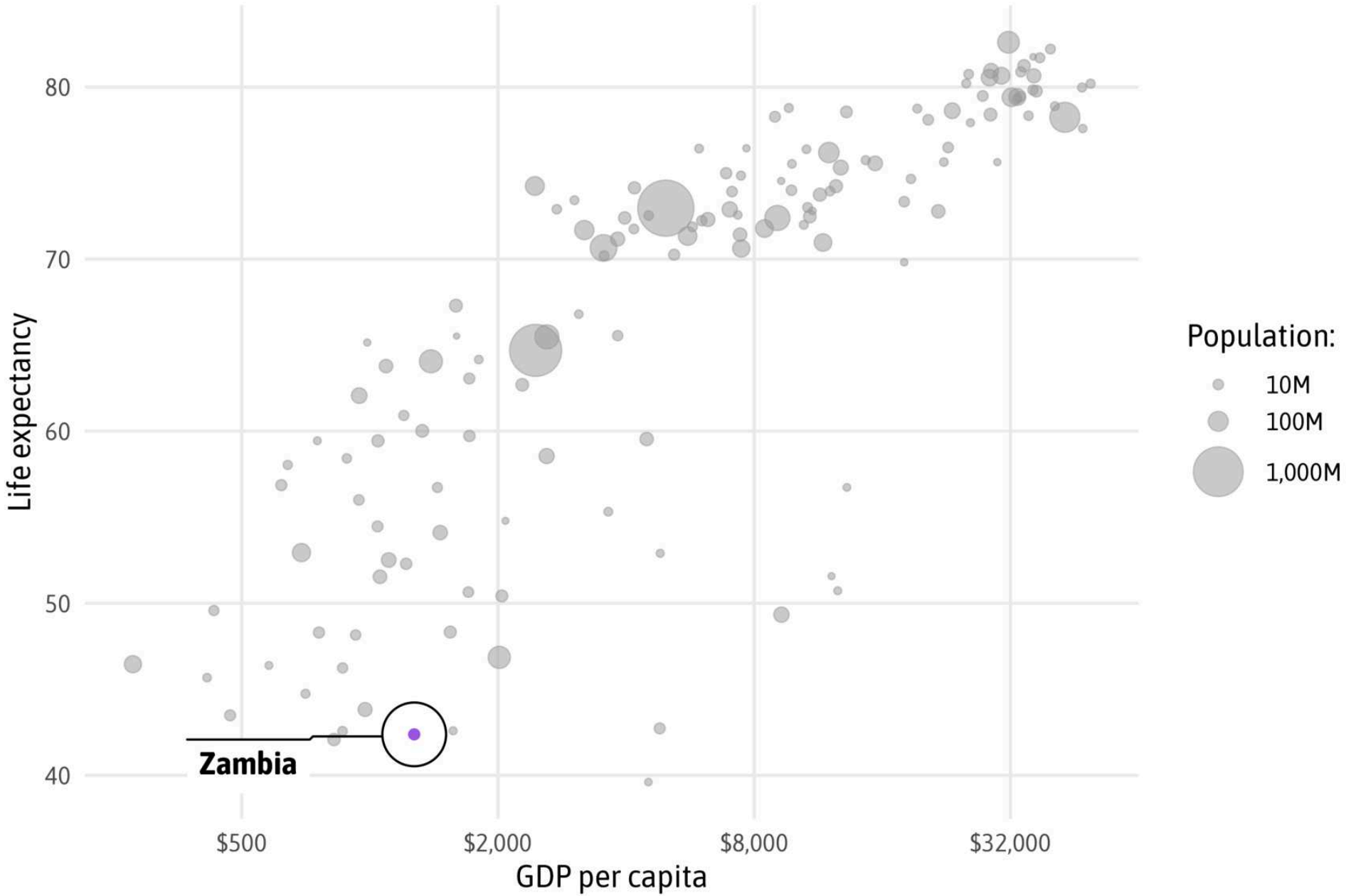
```
draw_gp_country("Germany")
```



```
draw_gp_country("India")
```



```
1  draw_gp_country("Zambia")
```



APPENDIX

Cédric Scherer Data Visualization & Information Design

Working with Aspect Rations

Cédric Scherer Data Visualization & Information Design

How to Work with Aspect Ratios

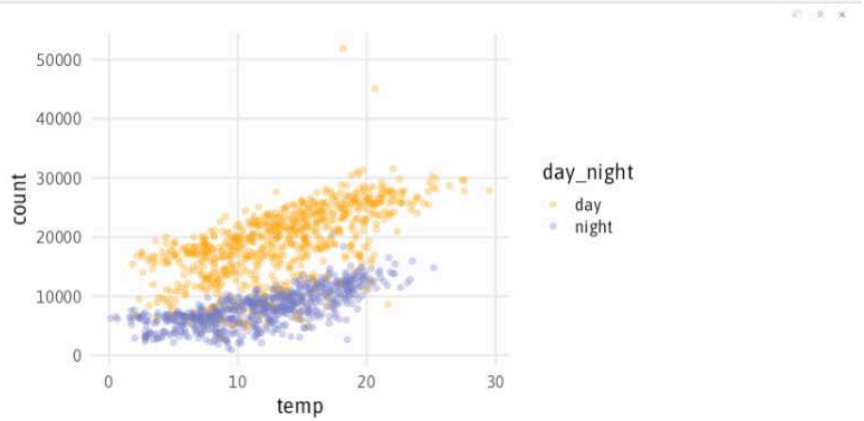
- don't rely on the Rstudio viewer pane!
- once you have a *"it's getting close"* prototype, settle on a plot size

- **Approach 1:** save the file and inspect it—go back to your IDE—repeat
  - tedious and time-consuming...

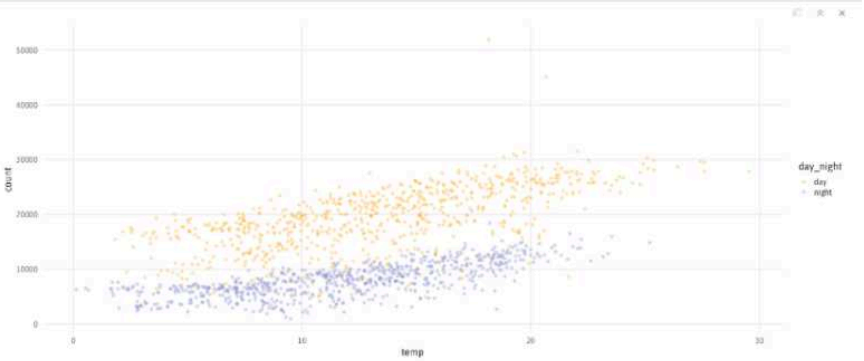
- **Approach 2:** use a qmd or rmd with inline output and chunk settings
  - set `fig-width`/`fig.width` and `fig-width`/`fig.width` as chunk option

Setting Plot Sizes in Qmd and Rmd

````
```{r}
#| label: plot-a
g
```
````




````
```{r, fig.width=20, fig.height=8}
#| label: plot-b
g
```
````



How to Work with Aspect Ratios

- don't rely on the Rstudio viewer pane!
- once you have a *"it's getting close"* prototype, settle on a plot size

- **Approach 1:** save the file and inspect it—go back to your IDE—repeat
  - tedious and time-consuming...

- **Approach 2:** use a qmd or rmd with inline output and chunk settings
  - set `fig.width` and `fig.height` per chunk or globally

- **Approach 3:** use our {camcorder} package
  - saves output from all `ggplot()` calls and displays it in the viewer pane

Setting Plot Sizes via {camcorder}

aspect-ratios-camcorder.qmd

Render on Save | Render | Run | Source | Visual | Outline

```yaml
---
title: "Working with Aspect Ratios: Camcorder Examples"
editor: source
---
```

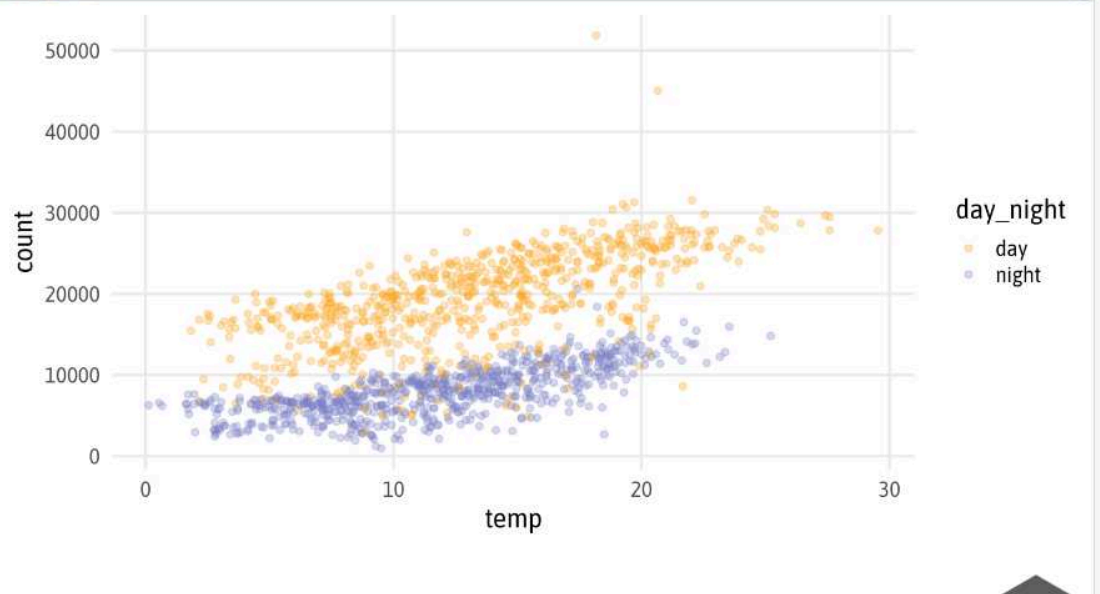
```r
library(ggplot2)
library(camcorder)

bikes <- readr::read_csv(here::here("data", "london-bikes.csv"),
                         col_types = "Dcfffiiiddddc")

gg_record(
  dir = here::here("temp"),
  device = "png",
  width = 10,
  height = 5,
  dpi = 600
)

ggplot(bikes, aes(x = temp, y = count, color = day_night)) +
  geom_point(alpha = .3, size = 2) +
  scale_color_manual(values = c(day = "#FFA200", night = "#757bc7")) +
  theme_minimal(base_size = 18, base_family = "Asap SemiCondensed") +
  theme(panel.grid.minor = element_blank())
```

Files | Plots | Packages | Help | Viewer | Presentation



Environment | History | Connections | Build | Git | Tutorial

Import Dataset | 293 MiB | R | Global Environment

Data

bikes	1454 obs. of 14 variables

Setting Plot Sizes via {camcorder}

aspect-ratios-camcorder.qmd

Render on Save | Render | Run

Source | Visual | Outline

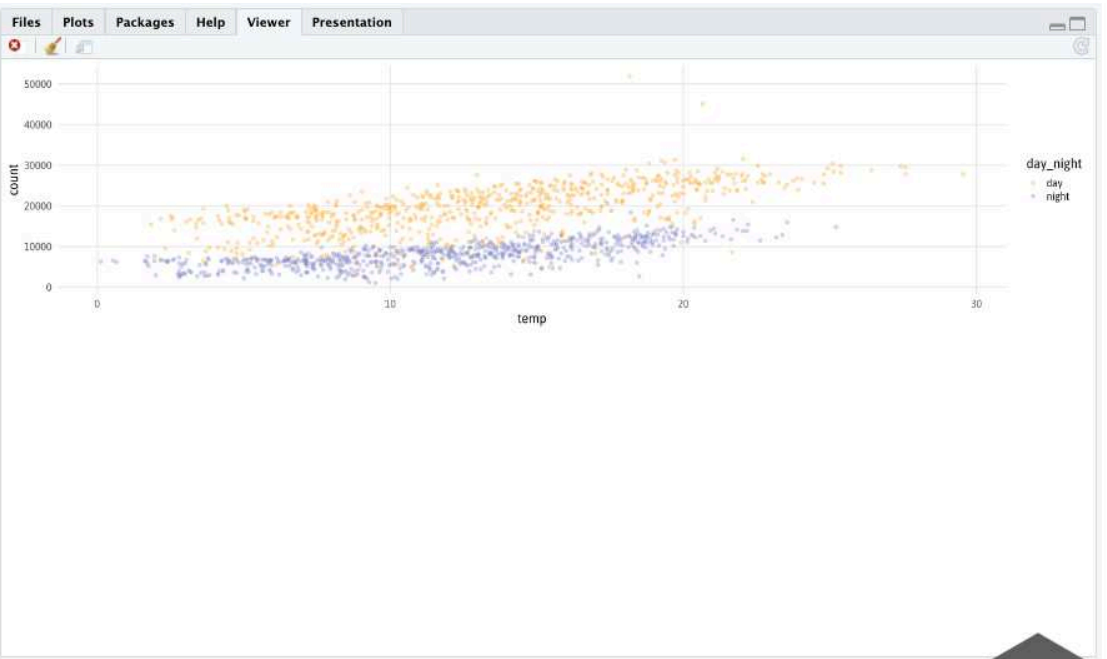
```
---
title: "Working with Aspect Ratios: Camcorder Examples"
editor: source
---

```{r}
library(ggplot2)
library(camcorder)

bikes <- readr::read_csv(here::here("data", "london-bikes.csv"),
                         col_types = "Dcfffilllddddc")

gg_record(
  dir = here::here("temp"),
  device = "png",
  width = 20,
  height = 5,
  dpi = 600
)

ggplot(bikes, aes(x = temp, y = count, color = day_night)) +
  geom_point(alpha = .3, size = 2) +
  scale_color_manual(values = c(day = "#FFA200", night = "#757bc7")) +
  theme_minimal(base_size = 18, base_family = "Asap SemiCondensed") +
  theme(panel.grid.minor = element_blank())
```
```



Environment | History | Connections | Build | Git | Tutorial

Import Dataset | 243 MiB

R | Global Environment

Data

bikes	1454 obs. of 14 variables

Setting Plot Sizes via {camcorder}

```
camcorder::gg_record(
  dir = here::here("temp"),  # path for plot files
  device = "png",            # device to use
  width = 10,                # figure width
  height = 5,                # figure height
  dpi = 600                  # plot resolution
)

g <- ggplot(fannie_sub, aes(x = orig_rate, y = dti, color = purpose)) +
  geom_point(alpha = .3, size = 2) +
  scale_color_manual(values = c(day = "#FFA200", night = "#757BC7")) +
  theme_minimal(base_size = 13, base_family = "Asap SemiCondensed") +
  theme(panel.grid.minor = element_blank())

g
```

Setting Plot Sizes via {camcorder}

```r
camcorder::gg_record(
  dir = here::here("temp"),  # path for plot files
  device = "png",            # device to use
  width = 10,                # figure width
  height = 5,                # figure height
  dpi = 600                  # plot resolution
)

g <- ggplot(fannie_sub, aes(x = orig_rate, y = dti, color = purpose)) +
  geom_point(alpha = .3, size = 2) +
  scale_color_manual(values = c(day = "#FFA200", night = "#757BC7")) +
  theme_minimal(base_size = 13, base_family = "Asap SemiCondensed") +
  theme(panel.grid.minor = element_blank())

g

camcorder::gg_resize_film(width = 20) # update figure width

g
```