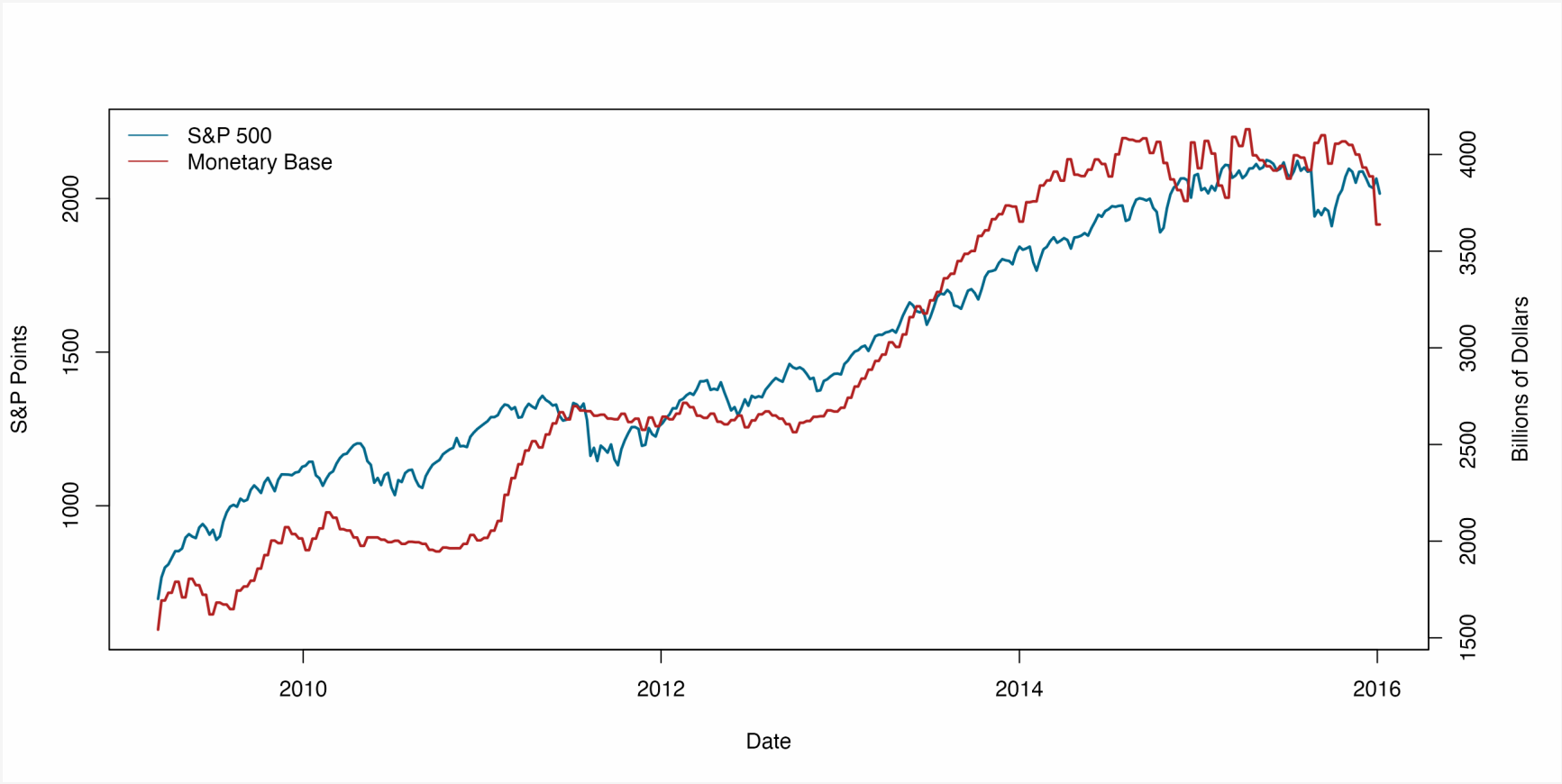
Some Case Studies

Data Visualization: Extra Material

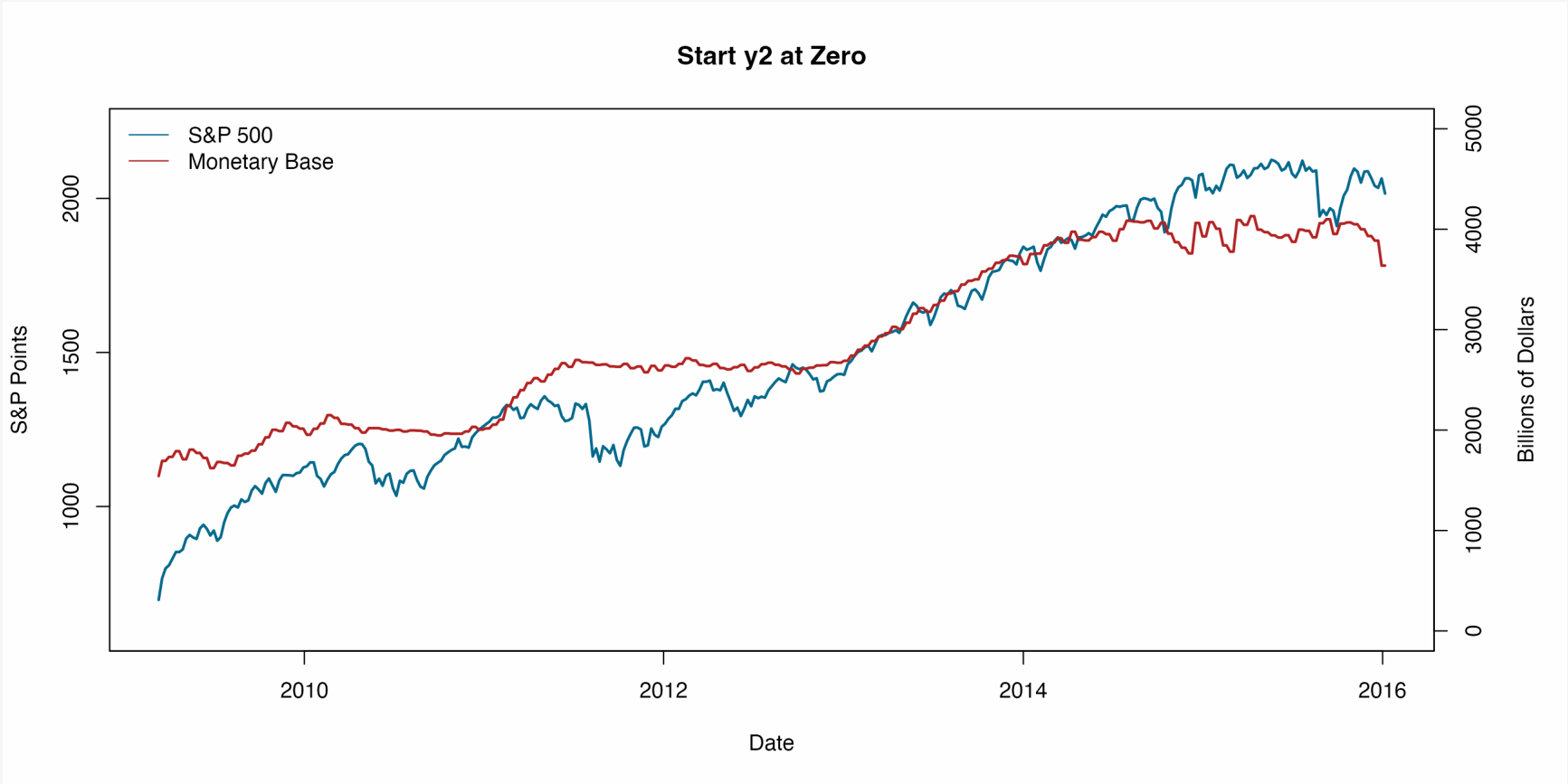
Kieran Healy
Duke University, May 2023

Working and Reworking

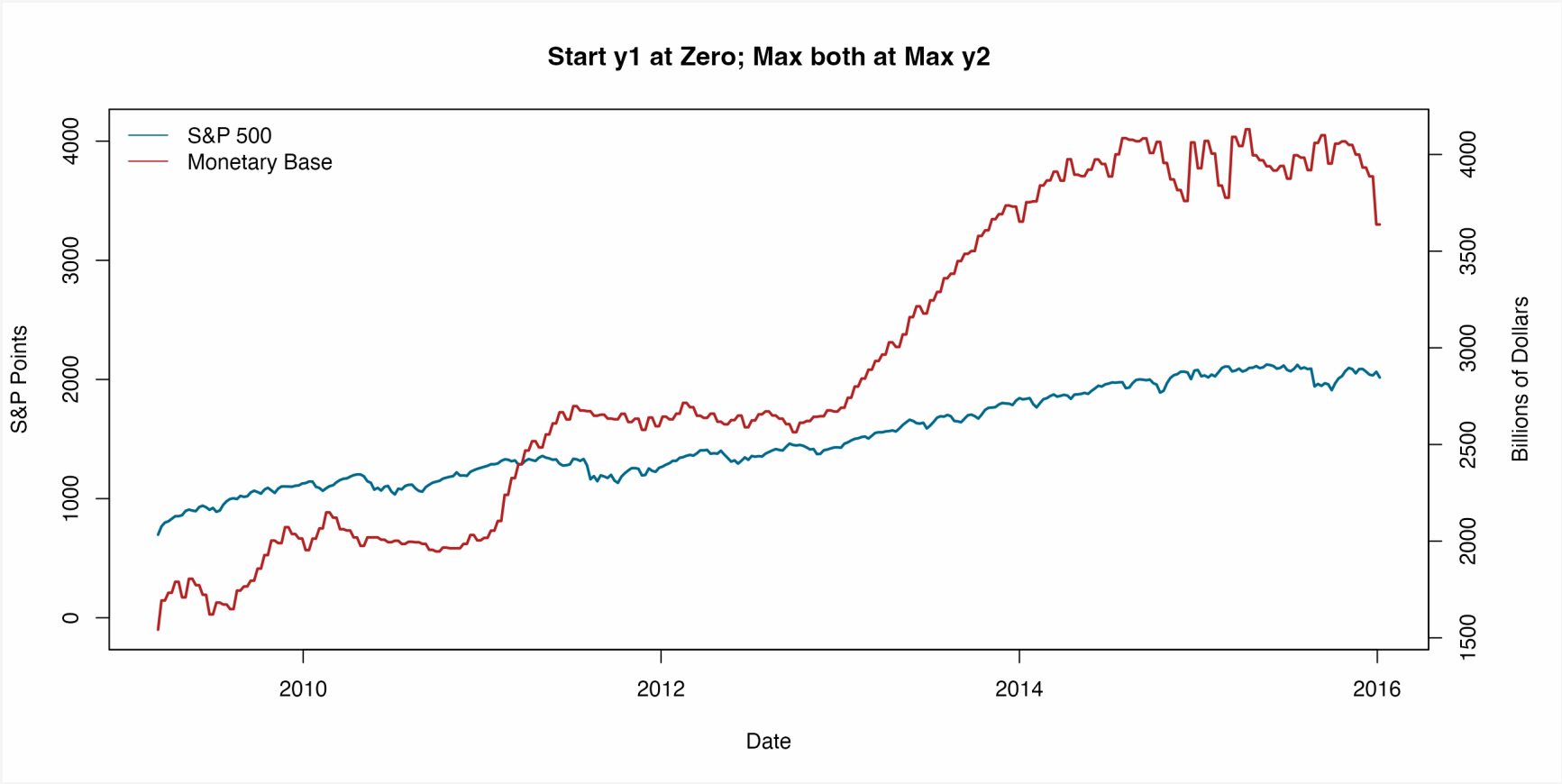
Two y-axes



Have we found the secret key to the stock market?



No, no we have not.


Start y1 at Zero; Max both at Max y2
S&P 500
Monetary Base
S&P Points
Billions of Dollars
0
1000
2000
3000
4000
1500
2000
2500
3000
3500
4000
2010
2012
2014
2016
Date

credit

What to do instead?

```
fredts ← as_tibble(fredts)
fredts
```

```
## # A tibble: 357 × 5
##    date       sp500 monbase sp500_i monbase_i
##    <date>     <dbl>   <int>   <dbl>     <dbl>
##  1 2009-03-11  697. 1542228    100       100
##  2 2009-03-18  767. 1693133    110.      110.
##  3 2009-03-25  799. 1693133    115.      110.
##  4 2009-04-01  809. 1733017    116.      112.
##  5 2009-04-08  831. 1733017    119.      112.
##  6 2009-04-15  852. 1789878    122.      116.
##  7 2009-04-22  852. 1789878    122.      116.
##  8 2009-04-29  861. 1709369    124.      111.
##  9 2009-05-06  896. 1709369    129.      111.
## 10 2009-05-13  908. 1805373    130.      117.
## # ℹ 347 more rows
```

Pivot the data

```
fredts
```

```
## # A tibble: 357 × 5
##    date       sp500 monbase sp500_i monbase_i
##    <date>     <dbl>   <int>   <dbl>     <dbl>
##  1 2009-03-11  697. 1542228    100        100
##  2 2009-03-18  767. 1693133    110.       110.
##  3 2009-03-25  799. 1693133    115.       110.
##  4 2009-04-01  809. 1733017    116.       112.
##  5 2009-04-08  831. 1733017    119.       112.
##  6 2009-04-15  852. 1789878    122.       116.
##  7 2009-04-22  852. 1789878    122.       116.
##  8 2009-04-29  861. 1709369    124.       111.
##  9 2009-05-06  896. 1709369    129.       111.
## 10 2009-05-13  908. 1805373    130.       117.
## # ℹ 347 more rows
```

Pivot the data

```
fredts ▷
  select(date, sp500_i, monbase_i)
```

```
## # A tibble: 357 × 3
##    date       sp500_i monbase_i
##    <date>       <dbl>     <dbl>
##  1 2009-03-11    100       100
##  2 2009-03-18    110.      110.
##  3 2009-03-25    115.      110.
##  4 2009-04-01    116.      112.
##  5 2009-04-08    119.      112.
##  6 2009-04-15    122.      116.
##  7 2009-04-22    122.      116.
##  8 2009-04-29    124.      111.
##  9 2009-05-06    129.      111.
## 10 2009-05-13    130.      117.
## # ℹ 347 more rows
```

Pivot the data

```
fredts ▷
  select(date, sp500_i, monbase_i)  ▷
  pivot_longer(sp500_i:monbase_i,
               names_to = "series",
               values_to = "score")
```

```
## # A tibble: 714 × 3
##    date       series    score
##    <date>     <chr>     <dbl>
##  1 2009-03-11 sp500_i    100
##  2 2009-03-11 monbase_i  100
##  3 2009-03-18 sp500_i    110.
##  4 2009-03-18 monbase_i  110.
##  5 2009-03-25 sp500_i    115.
##  6 2009-03-25 monbase_i  110.
##  7 2009-04-01 sp500_i    116.
##  8 2009-04-01 monbase_i  112.
##  9 2009-04-08 sp500_i    119.
## 10 2009-04-08 monbase_i  112.
## # ℹ 704 more rows
```

Pivot the data

```
fredts ▷
  select(date, sp500_i, monbase_i)  ▷
  pivot_longer(sp500_i:monbase_i,
               names_to = "series",
               values_to = "score") →
  fredts_m
```

Pivot the data

```
fredts ▷
  select(date, sp500_i, monbase_i)  ▷
  pivot_longer(sp500_i:monbase_i,
               names_to = "series",
               values_to = "score") →
  fredts_m
```

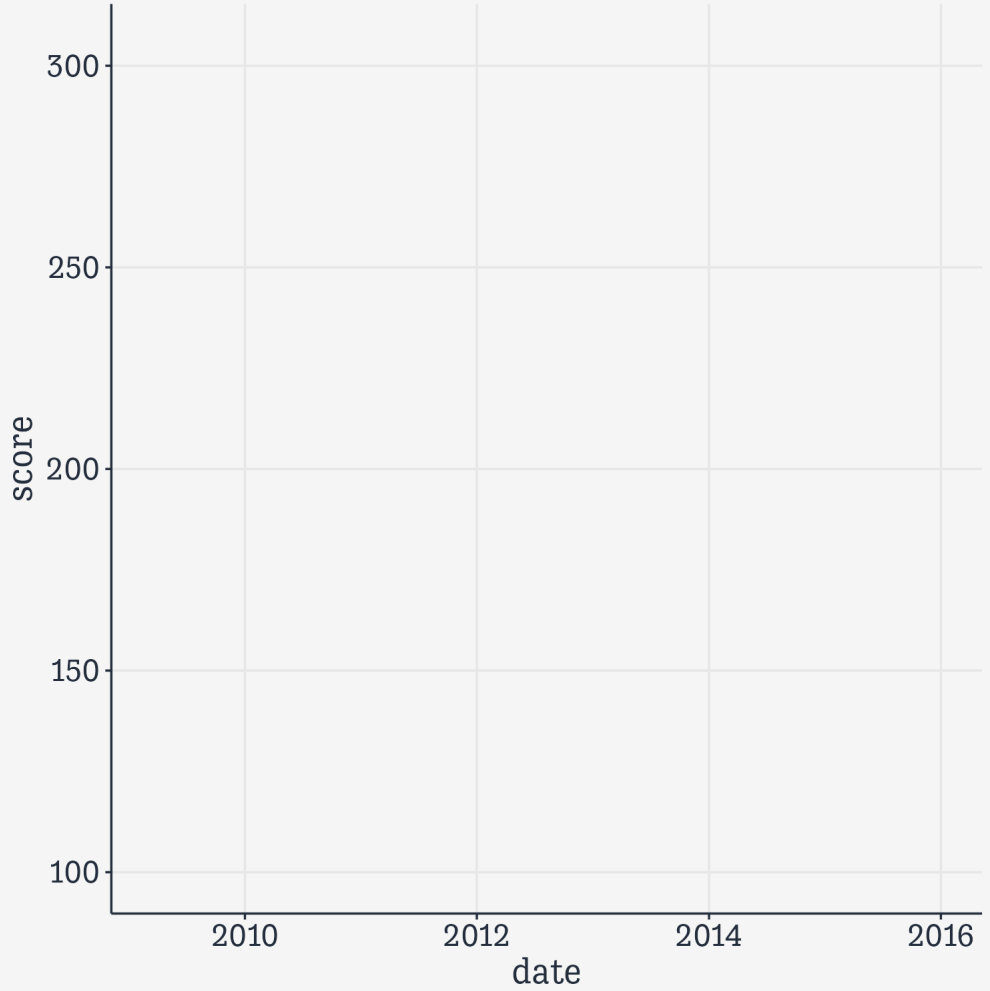
Make two plots

```
fredts_m
```

```
## # A tibble: 714 × 3
##    date       series    score
##    <date>     <chr>     <dbl>
##  1 2009-03-11 sp500_i    100
##  2 2009-03-11 monbase_i  100
##  3 2009-03-18 sp500_i    110.
##  4 2009-03-18 monbase_i  110.
##  5 2009-03-25 sp500_i    115.
##  6 2009-03-25 monbase_i  110.
##  7 2009-04-01 sp500_i    116.
##  8 2009-04-01 monbase_i  112.
##  9 2009-04-08 sp500_i    119.
## 10 2009-04-08 monbase_i  112.
## # ℹ 704 more rows
```

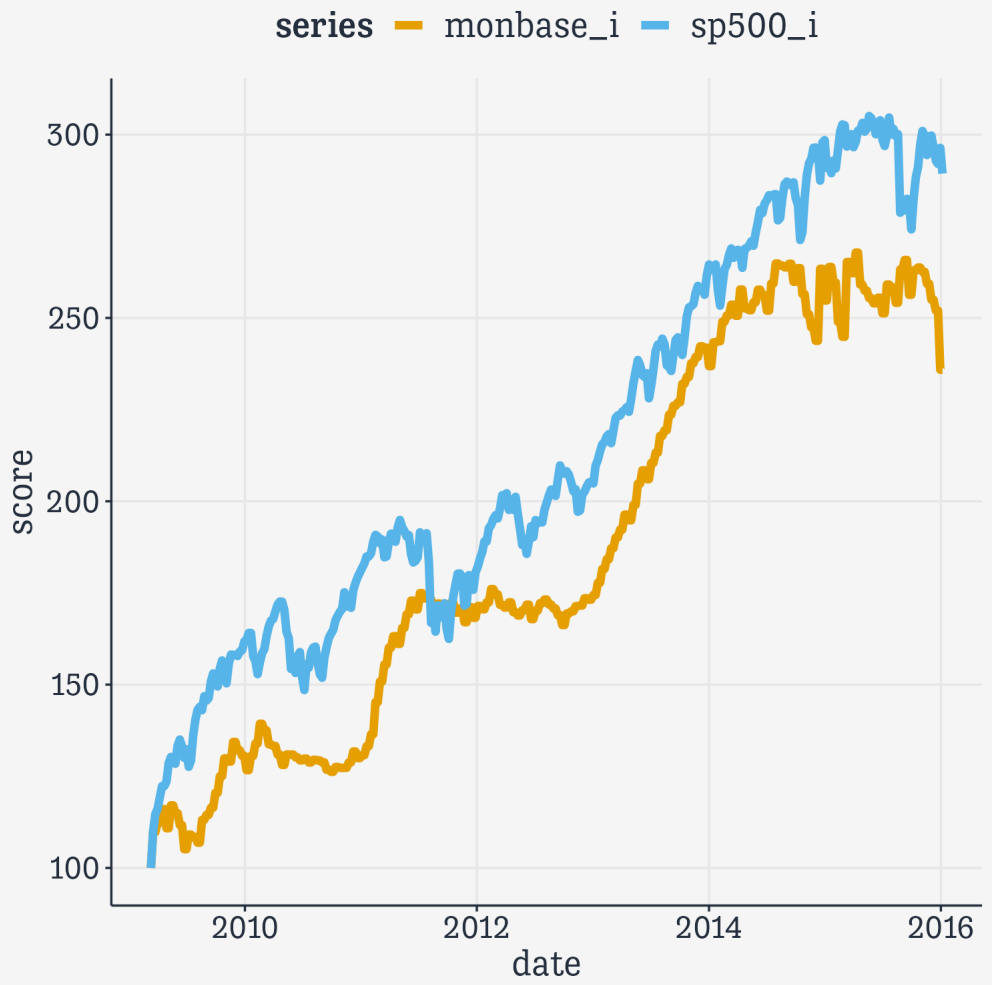
Make two plots

```
fredts_m ▷
  ggplot(mapping =
           aes(x = date,
               y = score,
               color = series))
```



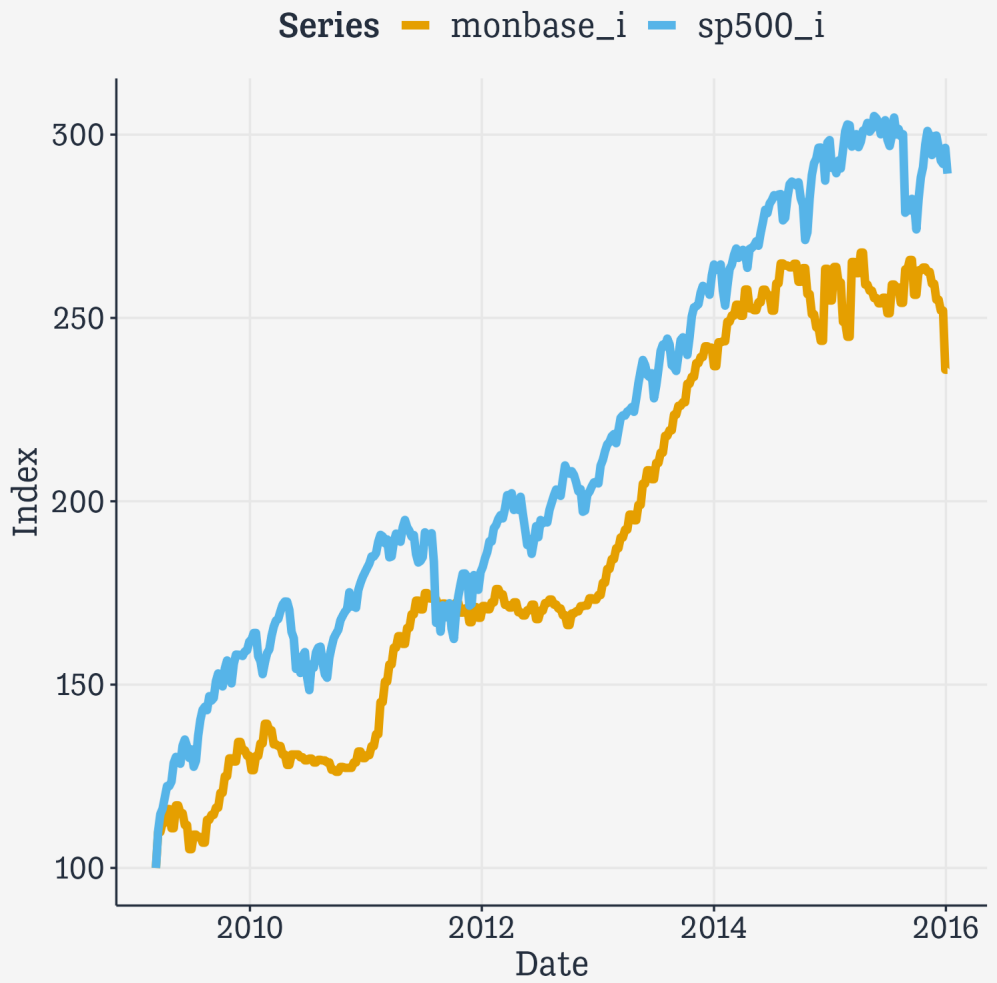
Make two plots

```
fredts_m ▷
  ggplot(mapping =
           aes(x = date,
               y = score,
               color = series)) +
  geom_line(linewidth = 2)
```



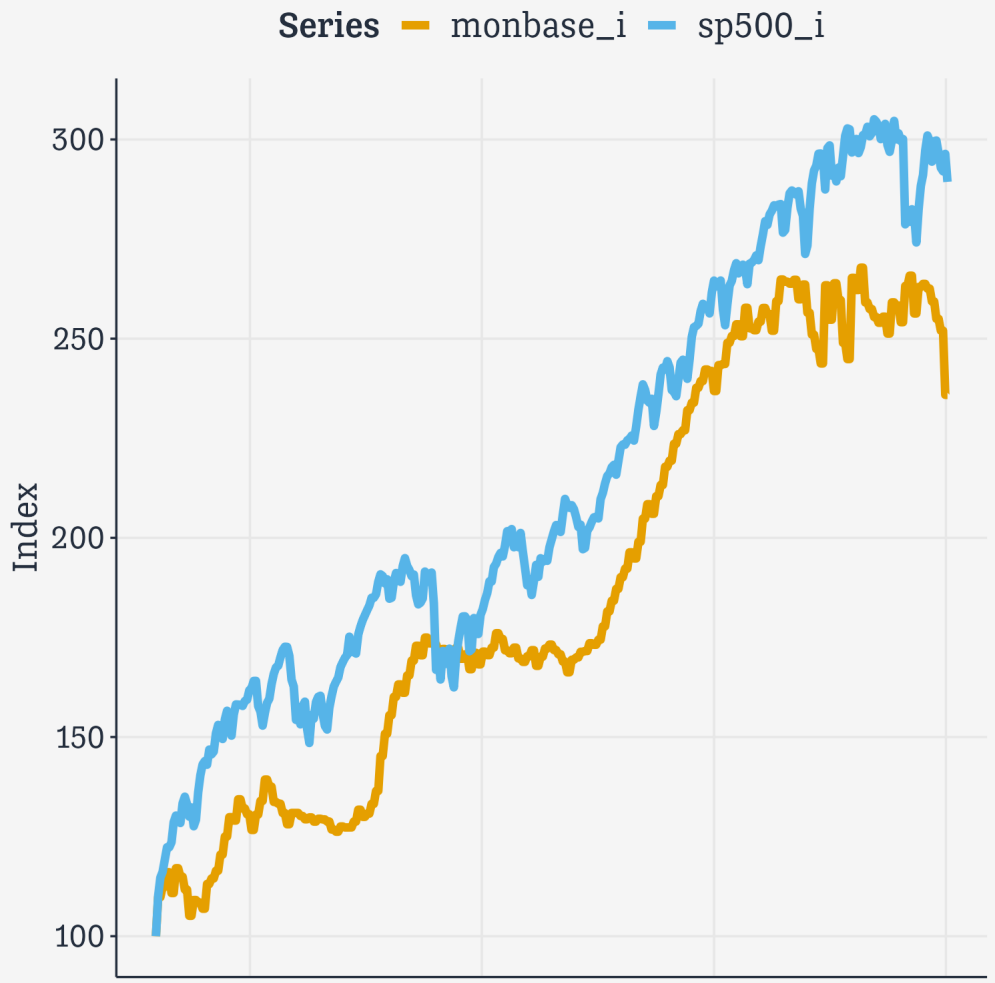
Make two plots

```
fredts_m ▷
  ggplot(mapping =
           aes(x = date,
               y = score,
               color = series)) +
  geom_line(linewidth = 2) +
  labs(x = "Date", y = "Index",
       color = "Series")
```



Make two plots

```
fredts_m ▷
  ggplot(mapping =
           aes(x = date,
               y = score,
               color = series)) +
  geom_line(linewidth = 2) +
  labs(x = "Date", y = "Index",
       color = "Series") +
  theme(axis.title.x = element_blank(),
        axis.text.x = element_blank(),
        axis.ticks.x = element_blank())
```



Make two plots

```
fredts_m ▷
  ggplot(mapping =
           aes(x = date,
               y = score,
               color = series)) +
  geom_line(linewidth = 2) +
  labs(x = "Date", y = "Index",
       color = "Series") +
  theme(axis.title.x = element_blank(),
        axis.text.x = element_blank(),
        axis.ticks.x = element_blank()) →
  p1
```

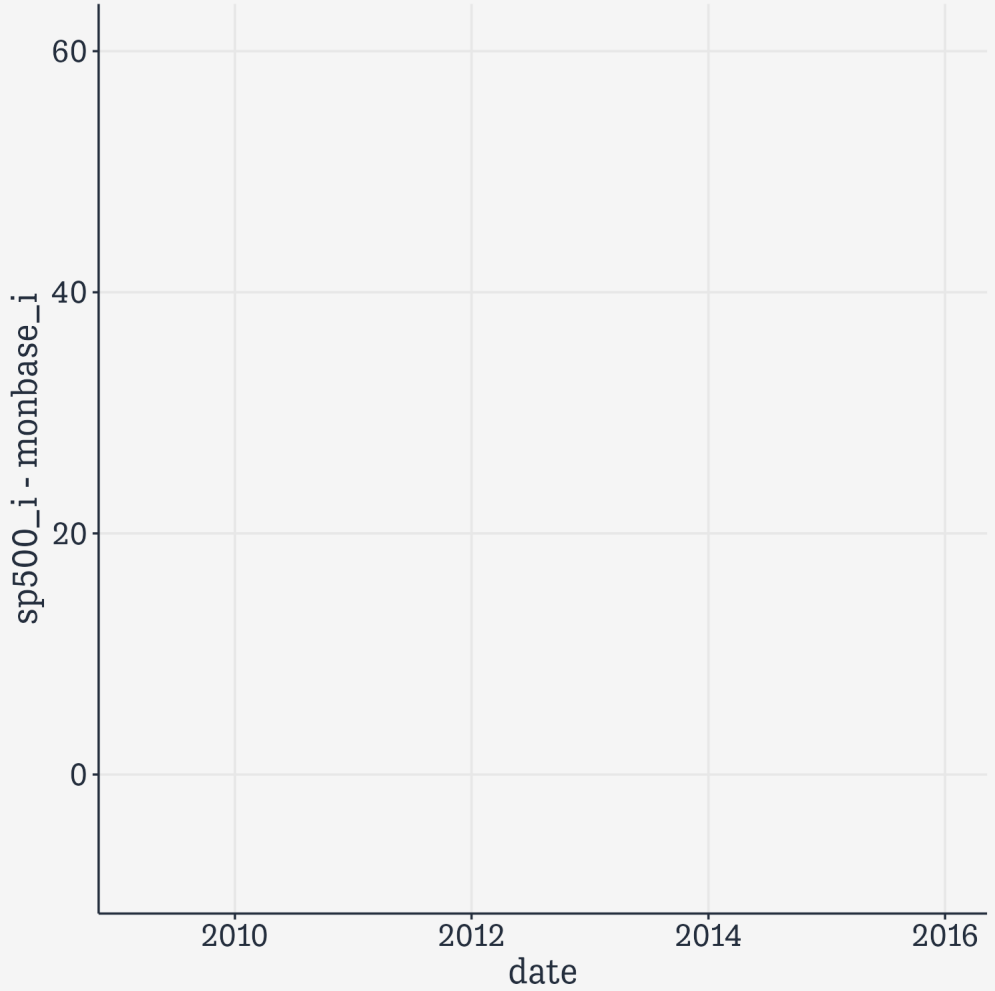
Make two plots

```
fredts_m ▷
  ggplot(mapping =
           aes(x = date,
               y = score,
               color = series)) +
  geom_line(linewidth = 2) +
  labs(x = "Date", y = "Index",
       color = "Series") +
  theme(axis.title.x = element_blank(),
        axis.text.x = element_blank(),
        axis.ticks.x = element_blank()) ⟶
  p1

# The original df
fredts
```

```
## # A tibble: 357 × 5
##    date       sp500 monbase sp500_i monbase_i
##    <date>     <dbl>   <int>   <dbl>     <dbl>
##  1 2009-03-11  697. 1542228    100       100
##  2 2009-03-18  767. 1693133    110.      110.
##  3 2009-03-25  799. 1693133    115.      110.
##  4 2009-04-01  809. 1733017    116.      112.
##  5 2009-04-08  831. 1733017    119.      112.
##  6 2009-04-15  852. 1789878    122.      116.
##  7 2009-04-22  852. 1789878    122.      116.
##  8 2009-04-29  861. 1709369    124.      111.
##  9 2009-05-06  896. 1709369    129.      111.
## 10 2009-05-13  908. 1805373    130.      117.
## # ℹ 347 more rows
```

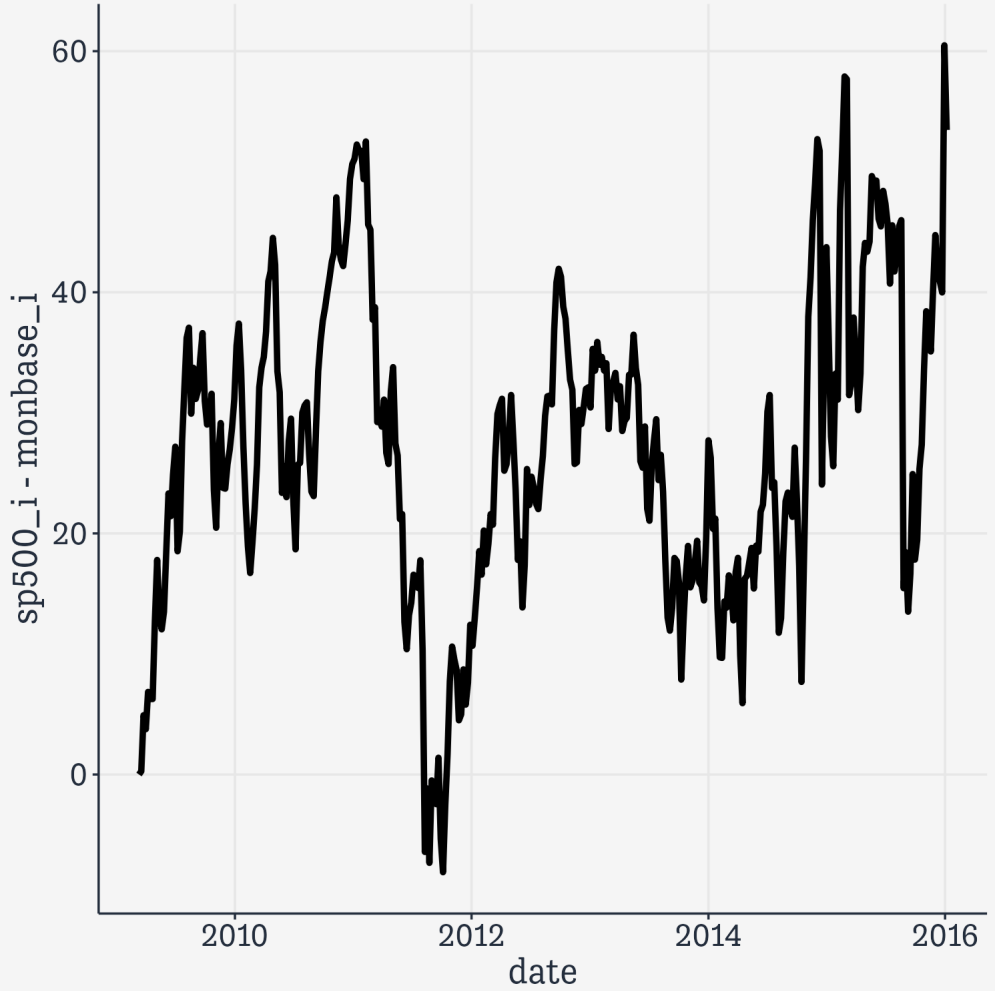
Make two plots

```
fredts_m ▷
  ggplot(mapping =
           aes(x = date,
               y = score,
               color = series)) +
  geom_line(linewidth = 2) +
  labs(x = "Date", y = "Index",
       color = "Series") +
  theme(axis.title.x = element_blank(),
        axis.text.x = element_blank(),
        axis.ticks.x = element_blank()) →
  p1

# The original df
fredts ▷
  ggplot(mapping =
           aes(x = date,
               y = sp500_i - monbase_i))
```



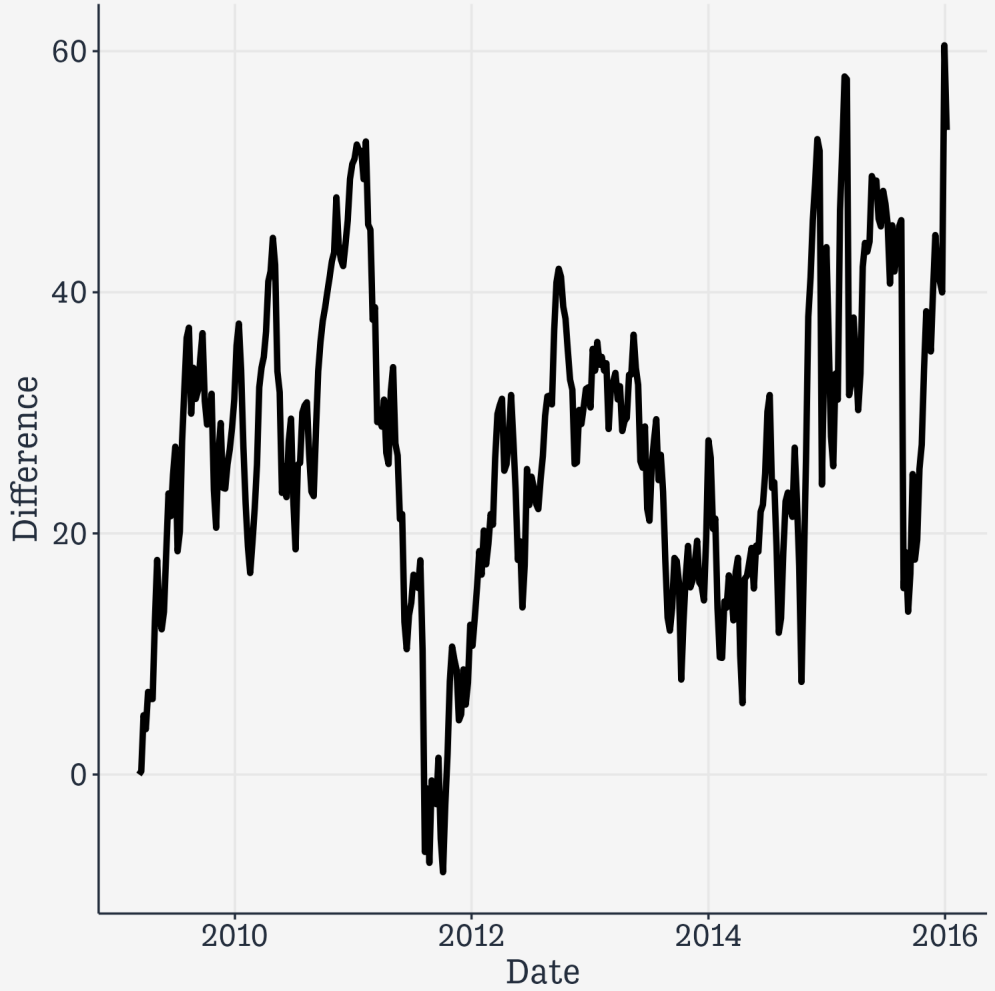
Make two plots

```
fredts_m ▷
  ggplot(mapping =
           aes(x = date,
               y = score,
               color = series)) +
  geom_line(linewidth = 2) +
  labs(x = "Date", y = "Index",
       color = "Series") +
  theme(axis.title.x = element_blank(),
        axis.text.x = element_blank(),
        axis.ticks.x = element_blank()) →
  p1

# The original df
fredts ▷
  ggplot(mapping =
          aes(x = date,
              y = sp500_i - monbase_i)) +
  geom_line(linewidth = 1.5)
```



Make two plots

```
fredts_m ▷
  ggplot(mapping =
           aes(x = date,
               y = score,
               color = series)) +
  geom_line(linewidth = 2) +
  labs(x = "Date", y = "Index",
       color = "Series") +
  theme(axis.title.x = element_blank(),
        axis.text.x = element_blank(),
        axis.ticks.x = element_blank()) →
  p1

# The original df
fredts ▷
  ggplot(mapping =
          aes(x = date,
              y = sp500_i - monbase_i)) +
  geom_line(linewidth = 1.5) +
  labs(x = "Date", y = "Difference")
```



Make two plots

```
fredts_m ▷
  ggplot(mapping =
           aes(x = date,
               y = score,
               color = series)) +
  geom_line(linewidth = 2) +
  labs(x = "Date", y = "Index",
       color = "Series") +
  theme(axis.title.x = element_blank(),
        axis.text.x = element_blank(),
        axis.ticks.x = element_blank()) ⟶
  p1

# The original df
fredts ▷
  ggplot(mapping =
          aes(x = date,
              y = sp500_i - monbase_i)) +
  geom_line(linewidth = 1.5) +
  labs(x = "Date", y = "Difference") ⟶
  p2
```

Make two plots

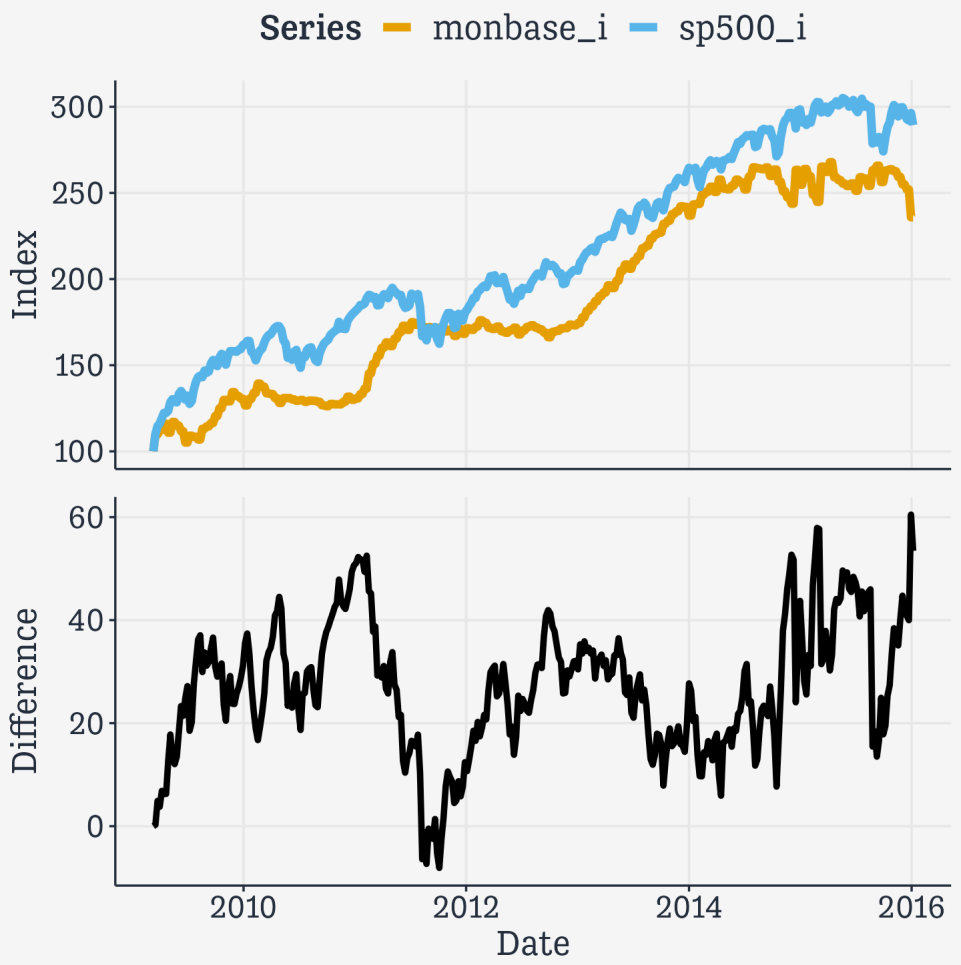
```
fredts_m ▷
  ggplot(mapping =
           aes(x = date,
               y = score,
               color = series)) +
  geom_line(linewidth = 2) +
  labs(x = "Date", y = "Index",
       color = "Series") +
  theme(axis.title.x = element_blank(),
        axis.text.x = element_blank(),
        axis.ticks.x = element_blank()) ⟶
  p1

# The original df
fredts ▷
  ggplot(mapping =
          aes(x = date,
              y = sp500_i - monbase_i)) +
  geom_line(linewidth = 1.5) +
  labs(x = "Date", y = "Difference") ⟶
  p2
```

Combine with patchwork

```
library(patchwork)
```

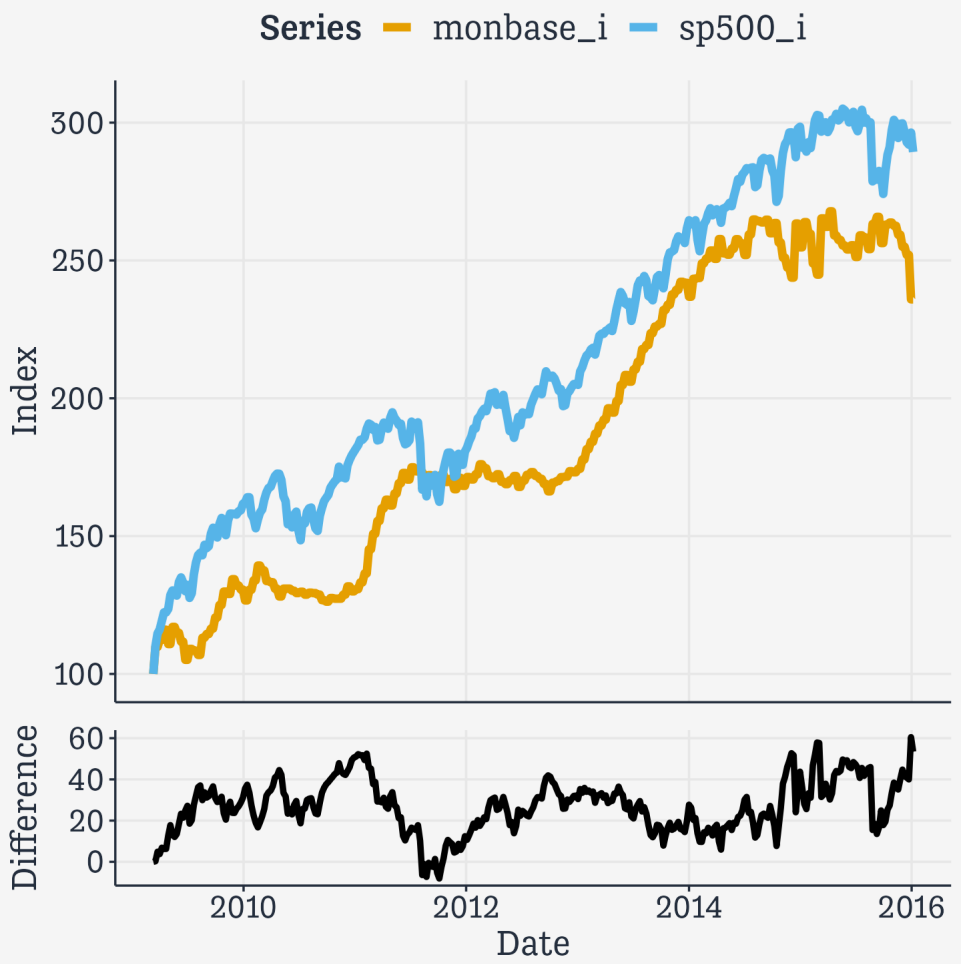
Combine with patchwork

```
library(patchwork)

(p1 / p2)
```



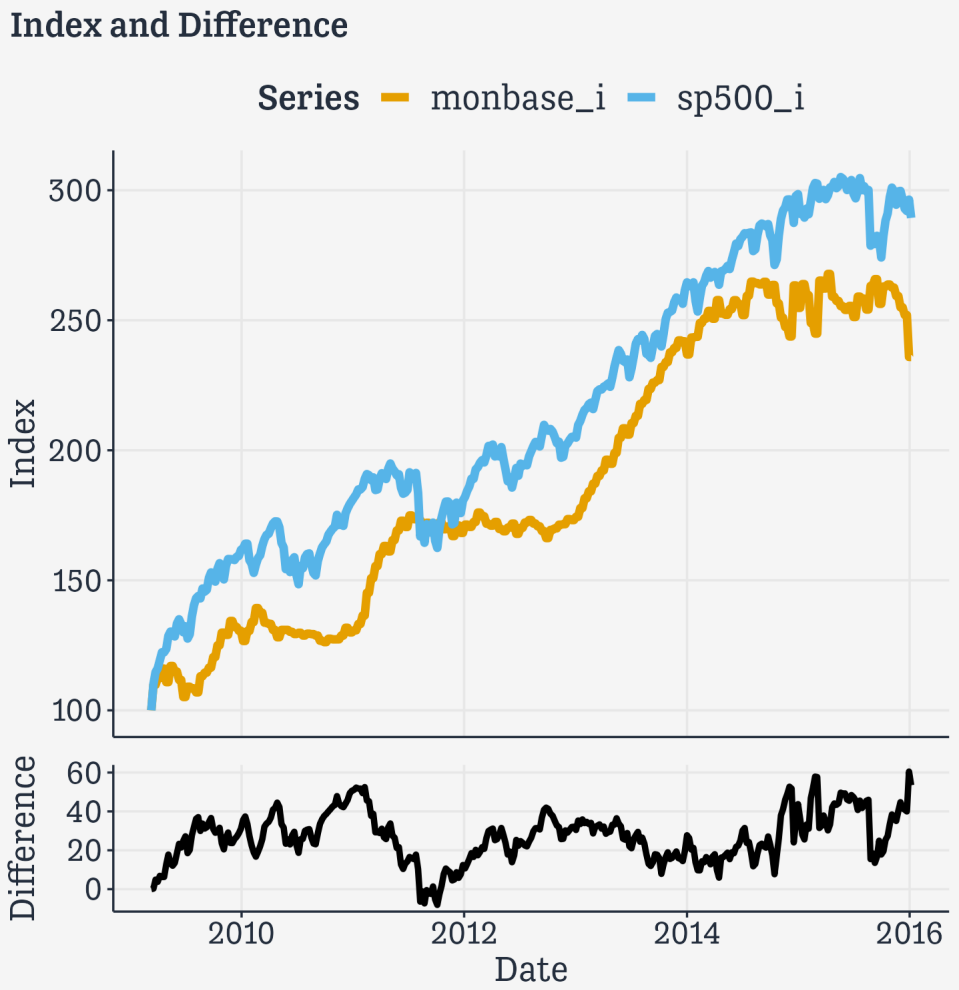
Combine with patchwork

```
library(patchwork)

(p1 / p2) +
  plot_layout(heights = c(4, 1))
```



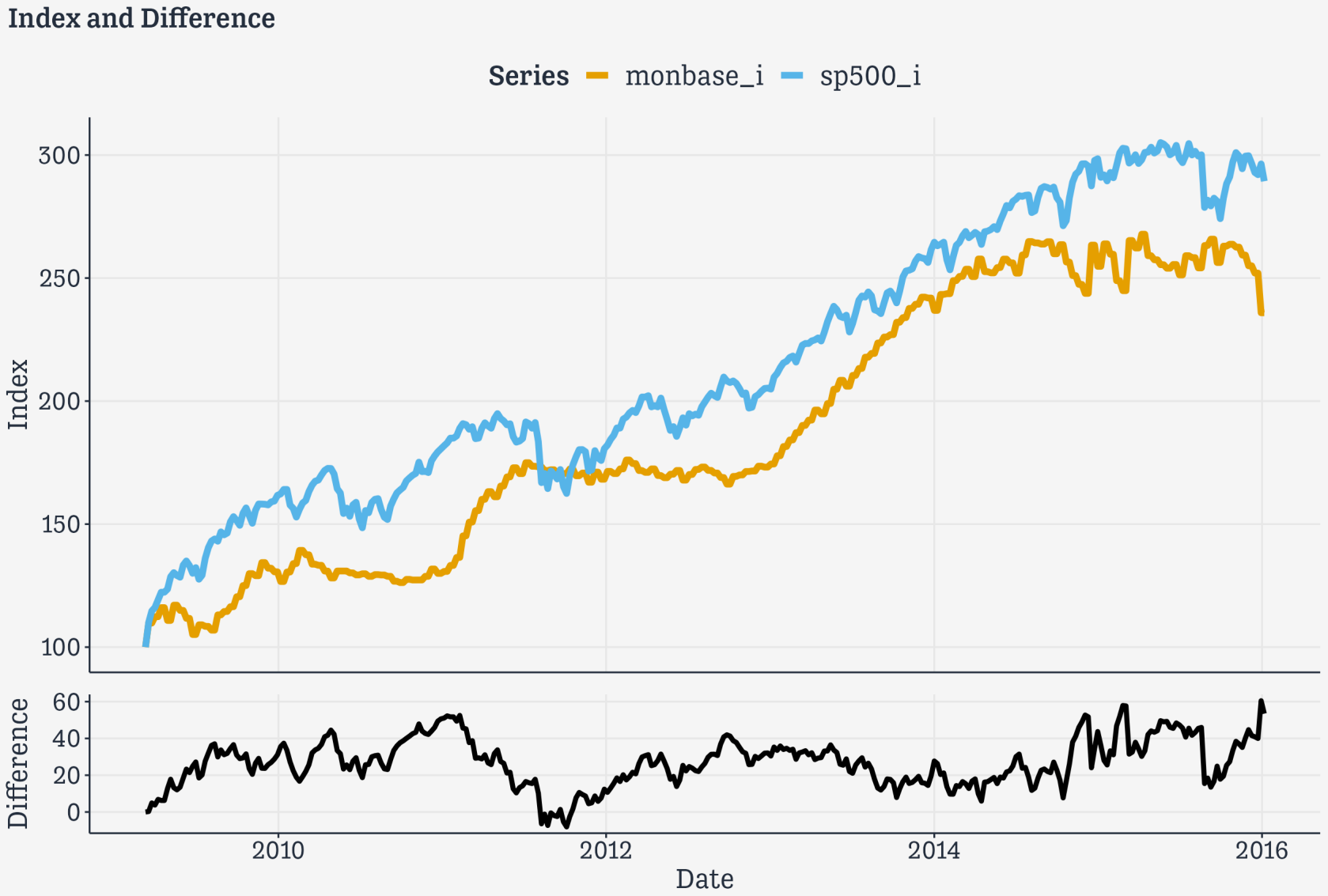
Combine with patchwork

```
library(patchwork)

(p1 / p2) +
  plot_layout(heights = c(4, 1)) +
  plot_annotation(title = "Index and Differ
```



Combine with patchwork

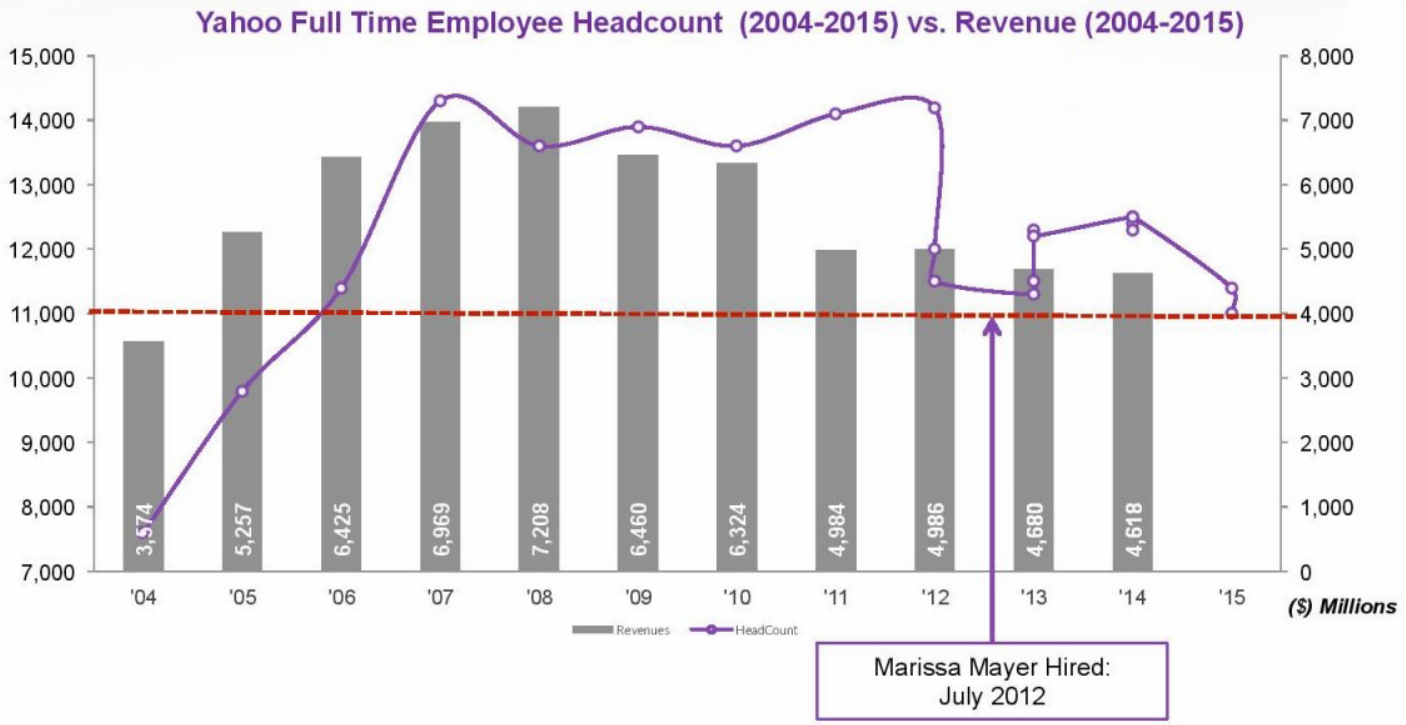
```
library(patchwork)

(p1 / p2) +
  plot_layout(heights = c(4, 1)) +
  plot_annotation(title = "Index and Differ
  p_patch
```



Patchwork plot.

Redrawing a bad slide

Yahoo's Headcount Still Excessively High Given Revenues:



Source: Company Filings (10K), Analyst calls

What can one say, really

The data

```
yahoo
```

```
## # A tibble: 12 × 4
##     Year Revenue Employees Mayer
##    <dbl>   <dbl>     <dbl> <chr>
##  1  2004    3574      7600 No
##  2  2005    5257      9800 No
##  3  2006    6425     11400 No
##  4  2007    6969     14300 No
##  5  2008    7208     13600 No
##  6  2009    6460     13900 No
##  7  2010    6324     13600 No
##  8  2011    4984     14100 No
##  9  2012    4986     12000 No
## 10  2012    4986     11500 Yes
## 11  2013    4680     12200 Yes
## 12  2014    4618     12500 Yes
```

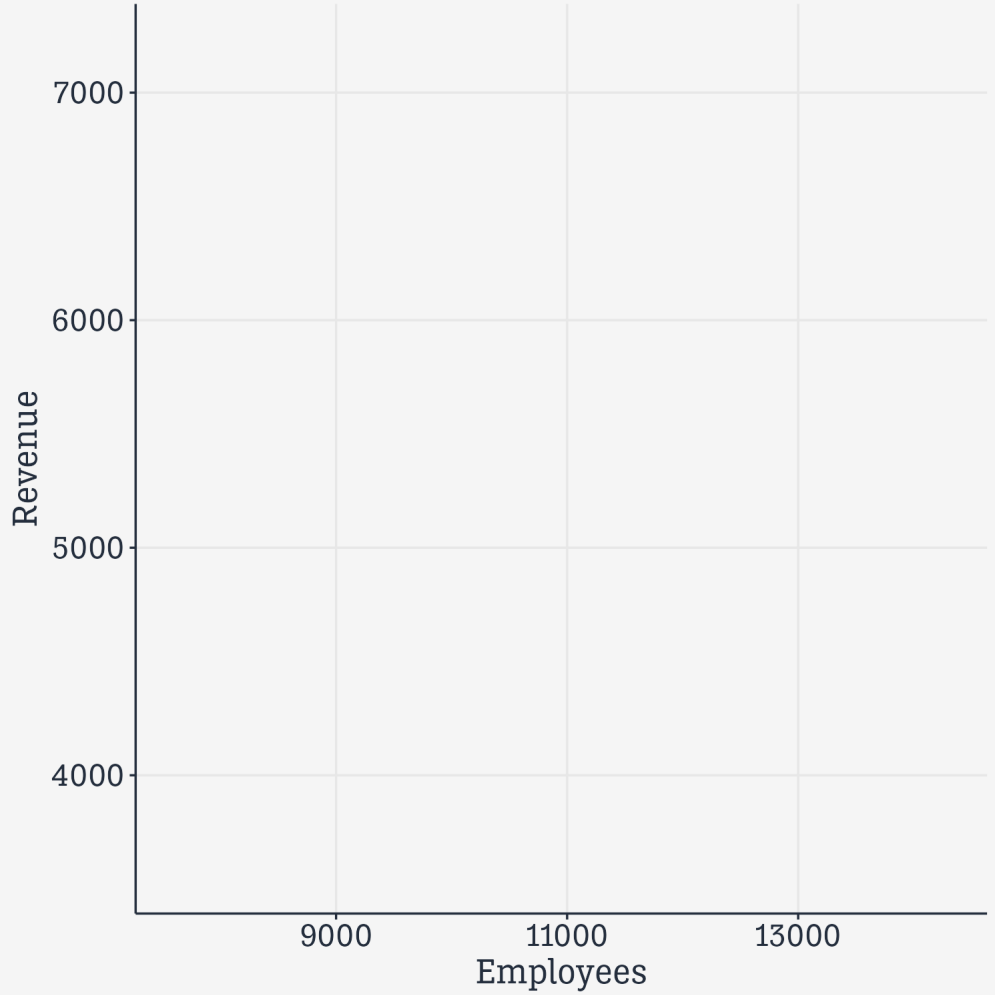
Option 1

```
yahoo
```

```
## # A tibble: 12 × 4
##     Year Revenue Employees Mayer
##    <dbl>   <dbl>     <dbl> <chr>
##  1  2004    3574      7600 No
##  2  2005    5257      9800 No
##  3  2006    6425     11400 No
##  4  2007    6969     14300 No
##  5  2008    7208     13600 No
##  6  2009    6460     13900 No
##  7  2010    6324     13600 No
##  8  2011    4984     14100 No
##  9  2012    4986     12000 No
## 10  2012    4986     11500 Yes
## 11  2013    4680     12200 Yes
## 12  2014    4618     12500 Yes
```

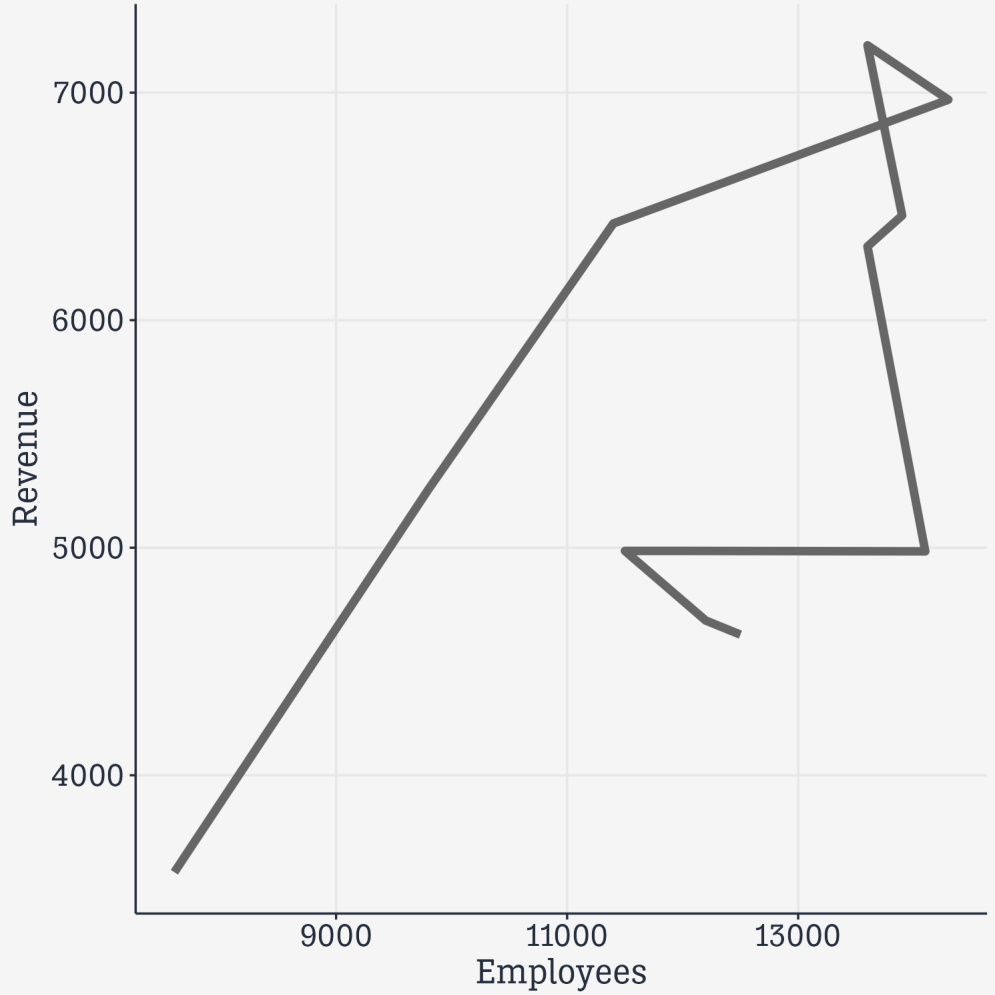
Option 1

```
yahoo ▷
  ggplot(mapping =
           aes(x = Employees,
               y = Revenue))
```



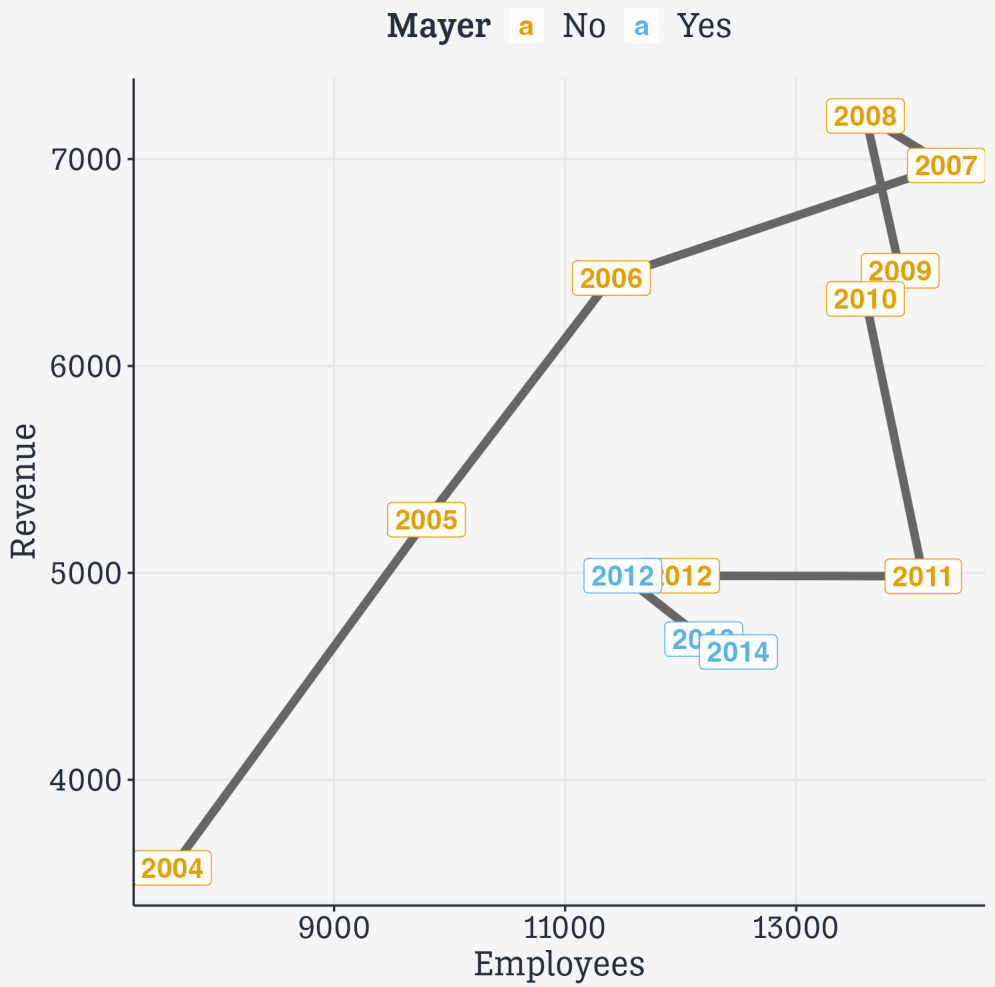
Option 1

```
yahoo ▷
  ggplot(mapping =
           aes(x = Employees,
               y = Revenue)) +
  geom_path(color = "gray40",
            linewidth = rel(2))
```



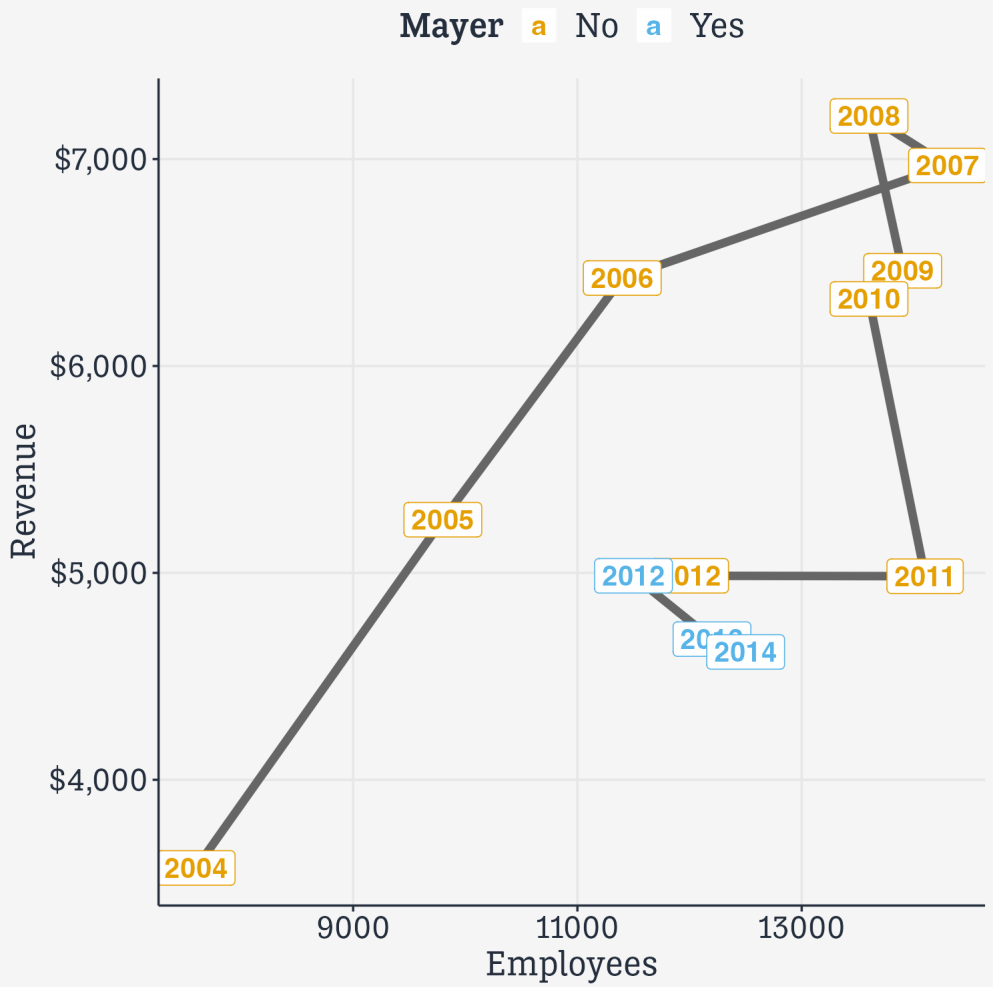
Option 1

```
yahoo ▷
  ggplot(mapping =
           aes(x = Employees,
               y = Revenue)) +
  geom_path(color = "gray40",
            linewidth = rel(2)) +
  geom_label(aes(color = Mayer,
                 label = Year),
             size = rel(5),
             fontface = "bold")
```



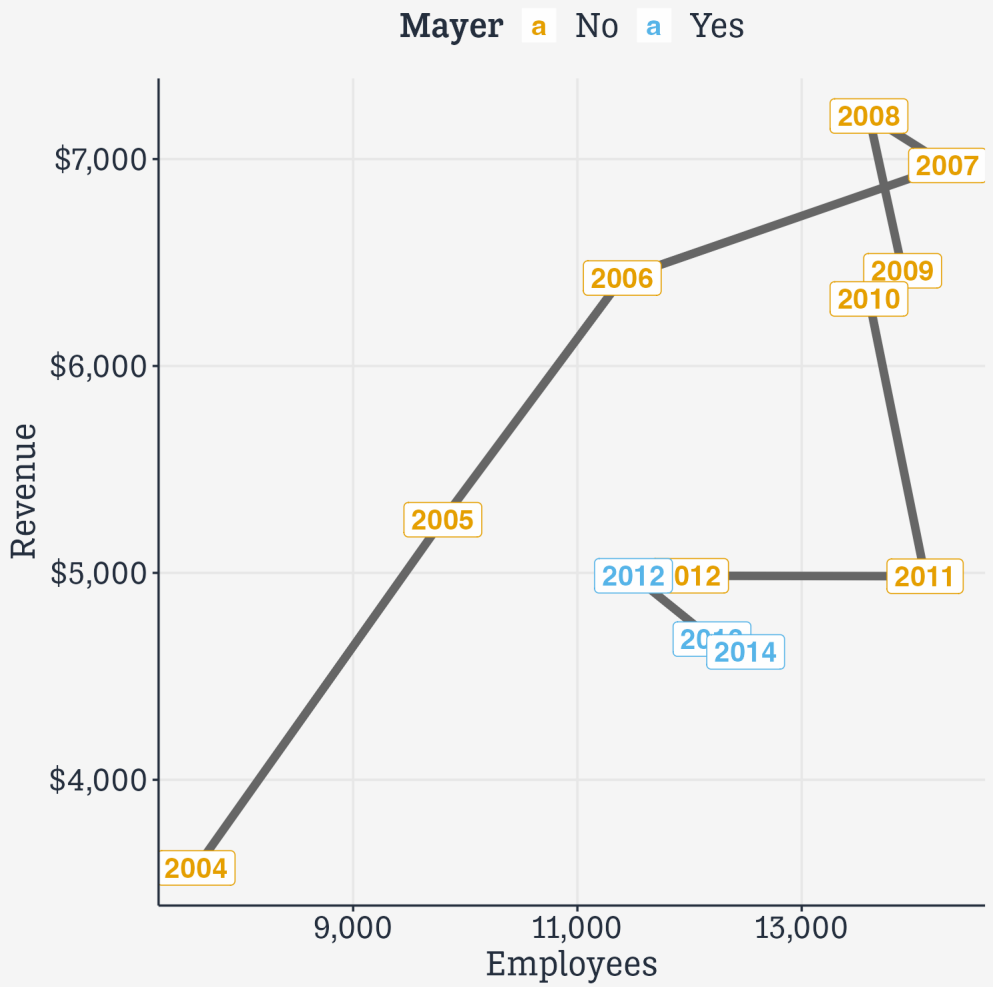
Option 1

```
yahoo ▷
  ggplot(mapping =
           aes(x = Employees,
               y = Revenue)) +
  geom_path(color = "gray40",
            linewidth = rel(2)) +
  geom_label(aes(color = Mayer,
                 label = Year),
             size = rel(5),
             fontface = "bold") +
  scale_y_continuous(labels = label_dollar(
```



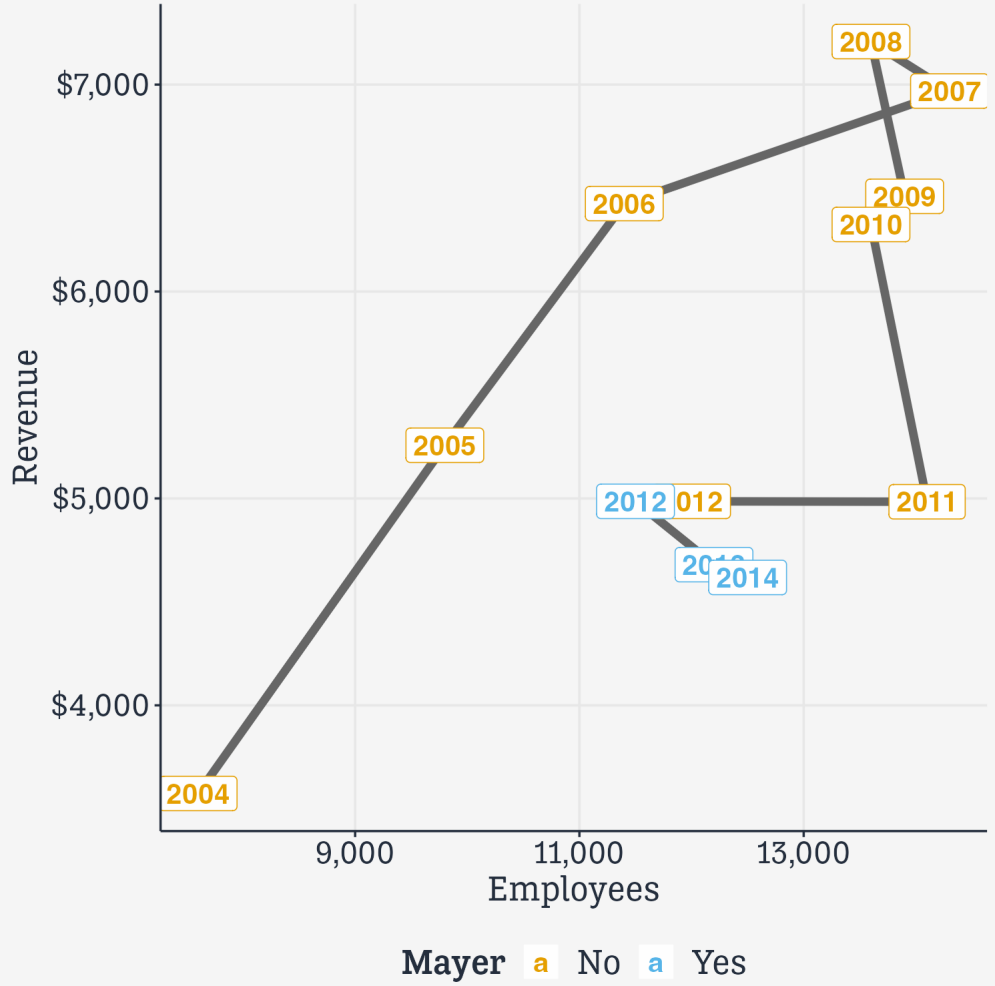
Option 1

```
yahoo ▷
  ggplot(mapping =
           aes(x = Employees,
               y = Revenue)) +
  geom_path(color = "gray40",
            linewidth = rel(2)) +
  geom_label(aes(color = Mayer,
                 label = Year),
             size = rel(5),
             fontface = "bold") +
  scale_y_continuous(labels = label_dollar(
  scale_x_continuous(labels = label_comma()
```



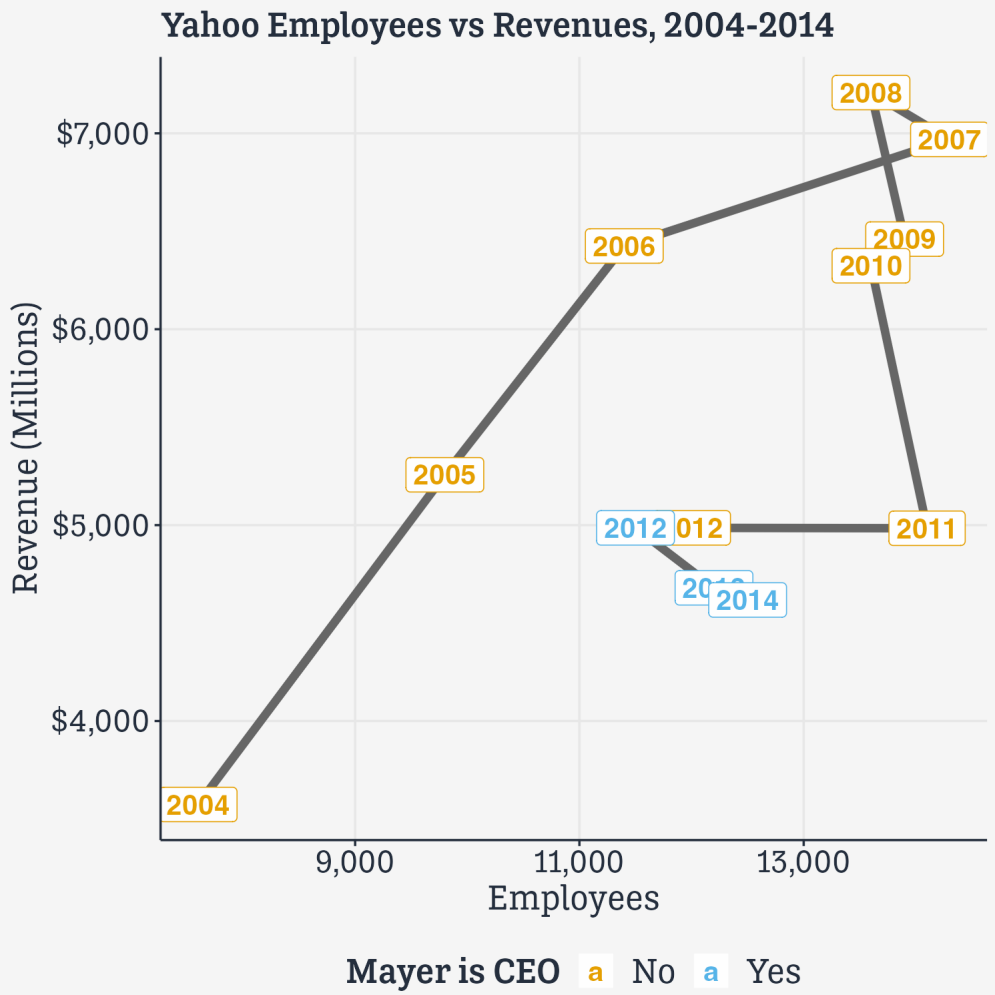
Option 1

```
yahoo ▷
  ggplot(mapping =
           aes(x = Employees,
               y = Revenue)) +
  geom_path(color = "gray40",
            linewidth = rel(2)) +
  geom_label(aes(color = Mayer,
                 label = Year),
             size = rel(5),
             fontface = "bold") +
  scale_y_continuous(labels = label_dollar(
  scale_x_continuous(labels = label_comma()
  theme(legend.position = "bottom")
```



Option 1

```
yahoo ▷
  ggplot(mapping =
           aes(x = Employees,
               y = Revenue)) +
  geom_path(color = "gray40",
            linewidth = rel(2)) +
  geom_label(aes(color = Mayer,
                 label = Year),
             size = rel(5),
             fontface = "bold") +
  scale_y_continuous(labels = label_dollar(
  scale_x_continuous(labels = label_comma()
  theme(legend.position = "bottom") +
  labs(color = "Mayer is CEO",
       x = "Employees", y = "Revenue (Milli
       title = "Yahoo Employees vs Revenues
```

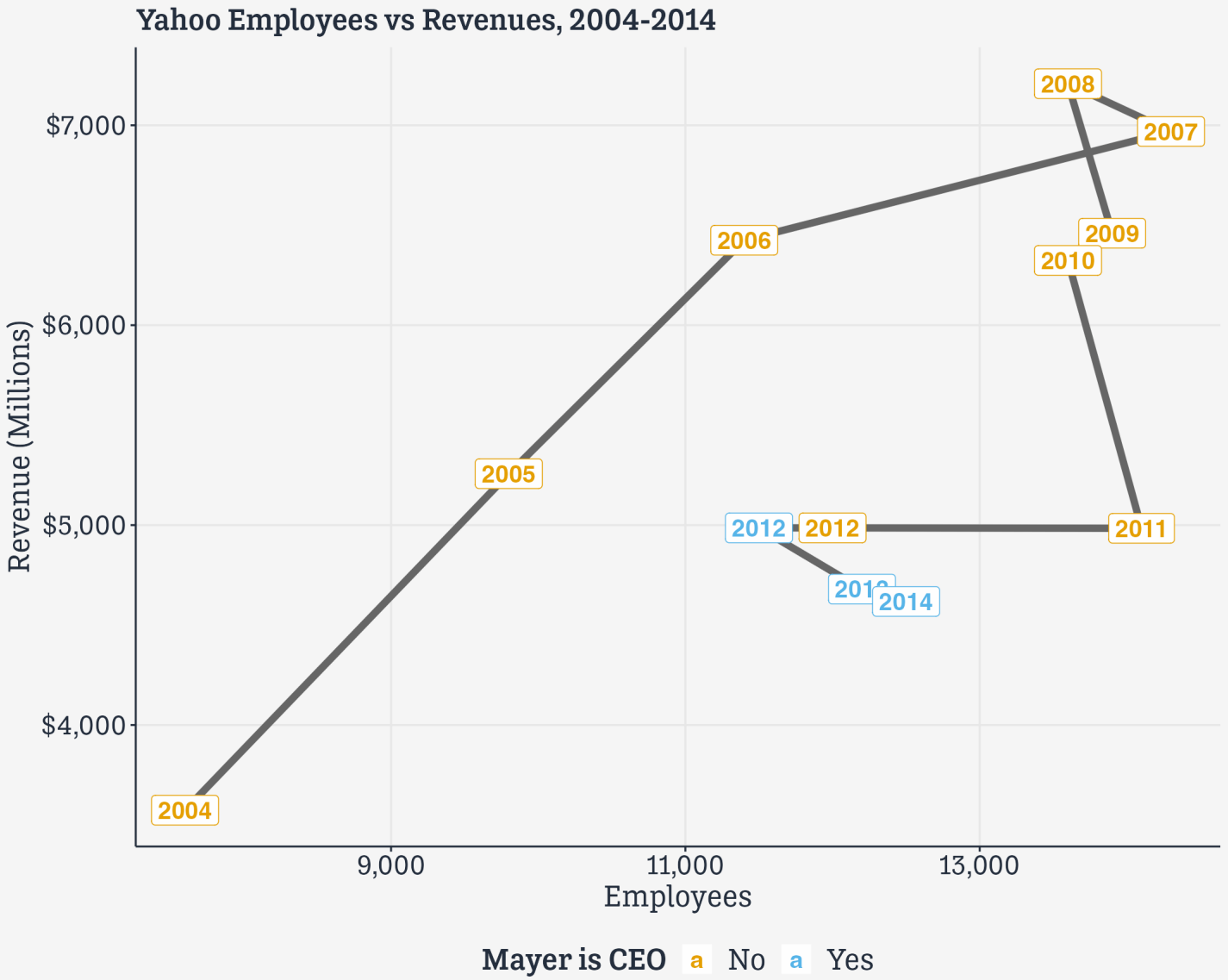


Option 1

```
yahoo ▷
  ggplot(mapping =
           aes(x = Employees,
               y = Revenue)) +
  geom_path(color = "gray40",
            linewidth = rel(2)) +
  geom_label(aes(color = Mayer,
                 label = Year),
             size = rel(5),
             fontface = "bold") +
  scale_y_continuous(labels = label_dollar(
  scale_x_continuous(labels = label_comma()
  theme(legend.position = "bottom") +
  labs(color = "Mayer is CEO",
       x = "Employees", y = "Revenue (Milli
       title = "Yahoo Employees vs Revenues
  yahoo1
```

Option 1

```
yahoo ▷
  ggplot(mapping =
           aes(x = Employees,
               y = Revenue)) +
  geom_path(color = "gray40",
            linewidth = rel(2)) +
  geom_label(aes(color = Mayer,
                 label = Year),
             size = rel(5),
             fontface = "bold") +
  scale_y_continuous(labels = label_dollar(
  scale_x_continuous(labels = label_comma()
  theme(legend.position = "bottom") +
  labs(color = "Mayer is CEO",
       x = "Employees", y = "Revenue (Milli
       title = "Yahoo Employees vs Revenues
  yahoo1
```



Redrawn with `geom_path()`

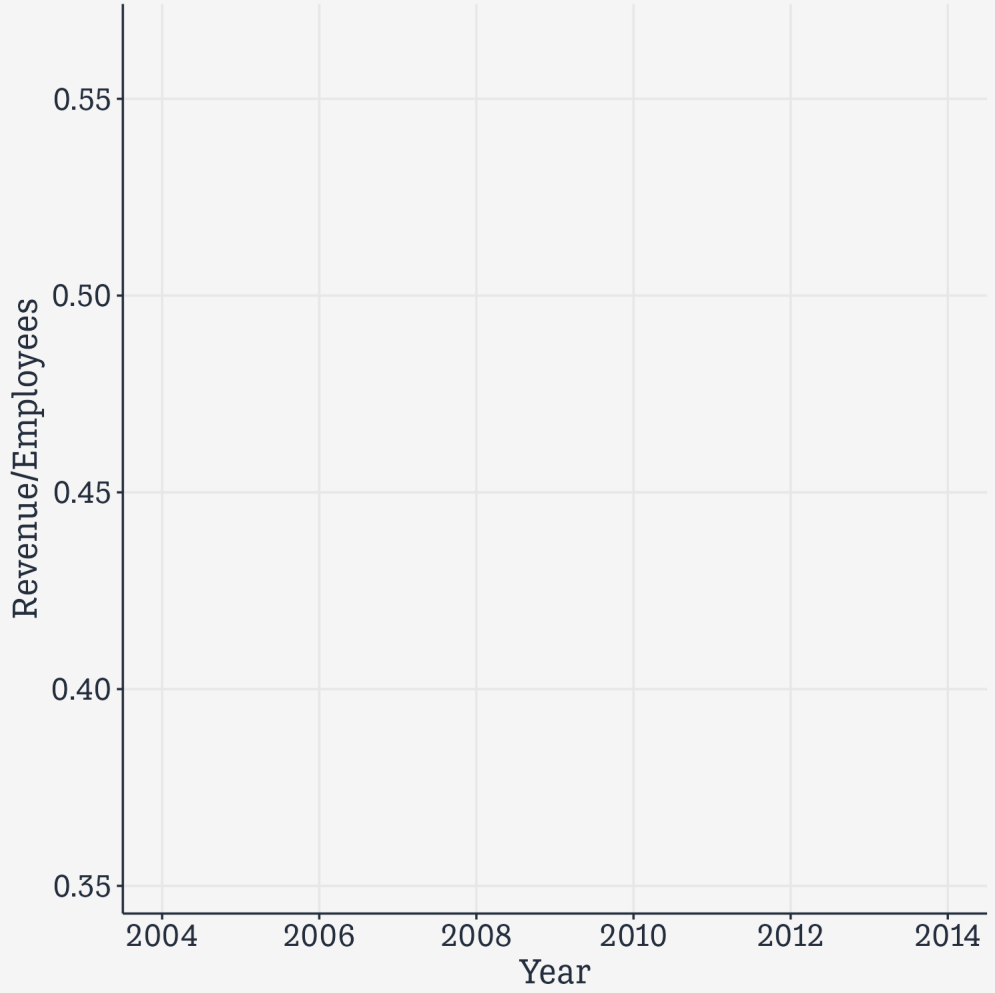
Alternatively ...

```
yahoo
```

```
## # A tibble: 12 × 4
##     Year Revenue Employees Mayer
##    <dbl>   <dbl>     <dbl> <chr>
##  1  2004    3574      7600 No
##  2  2005    5257      9800 No
##  3  2006    6425     11400 No
##  4  2007    6969     14300 No
##  5  2008    7208     13600 No
##  6  2009    6460     13900 No
##  7  2010    6324     13600 No
##  8  2011    4984     14100 No
##  9  2012    4986     12000 No
## 10  2012    4986     11500 Yes
## 11  2013    4680     12200 Yes
## 12  2014    4618     12500 Yes
```

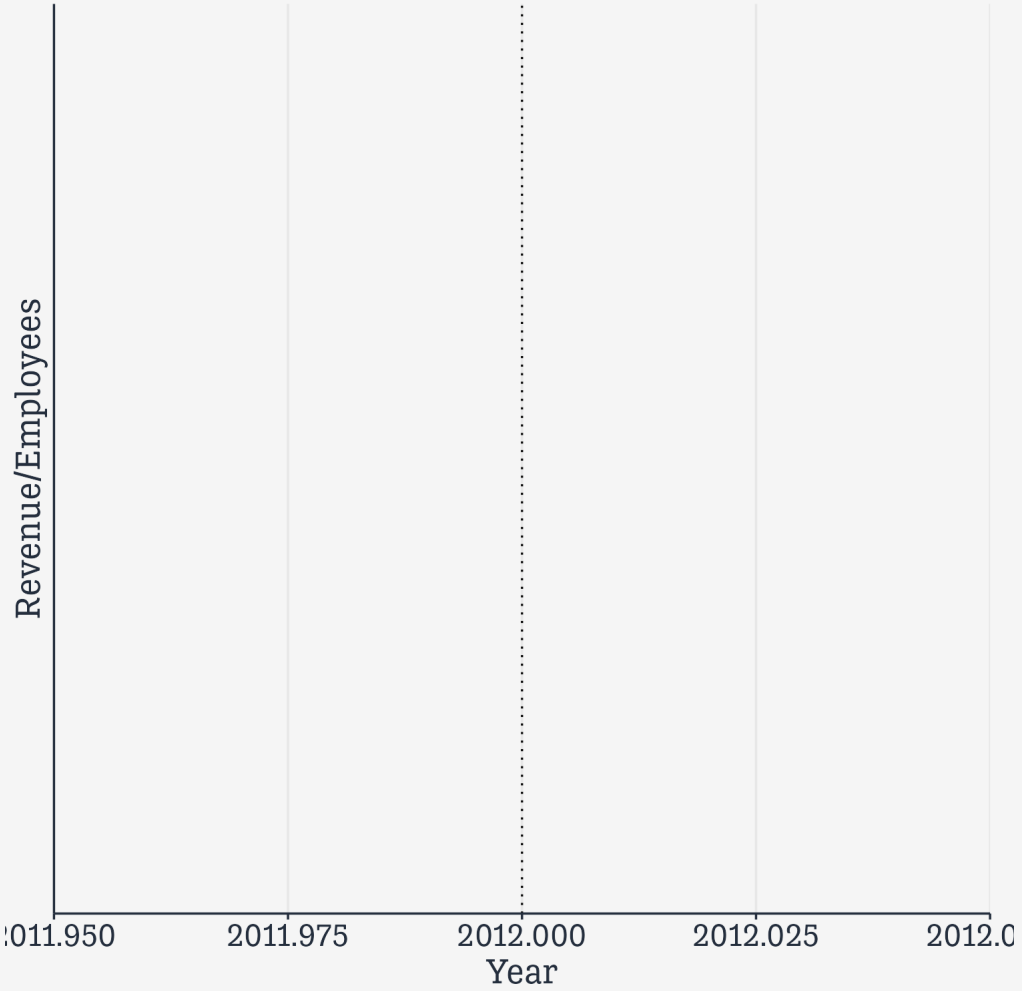
Alternatively ...

```
yahoo ▷
  ggplot(mapping =
           aes(x = Year,
               y = Revenue/Employees))
```



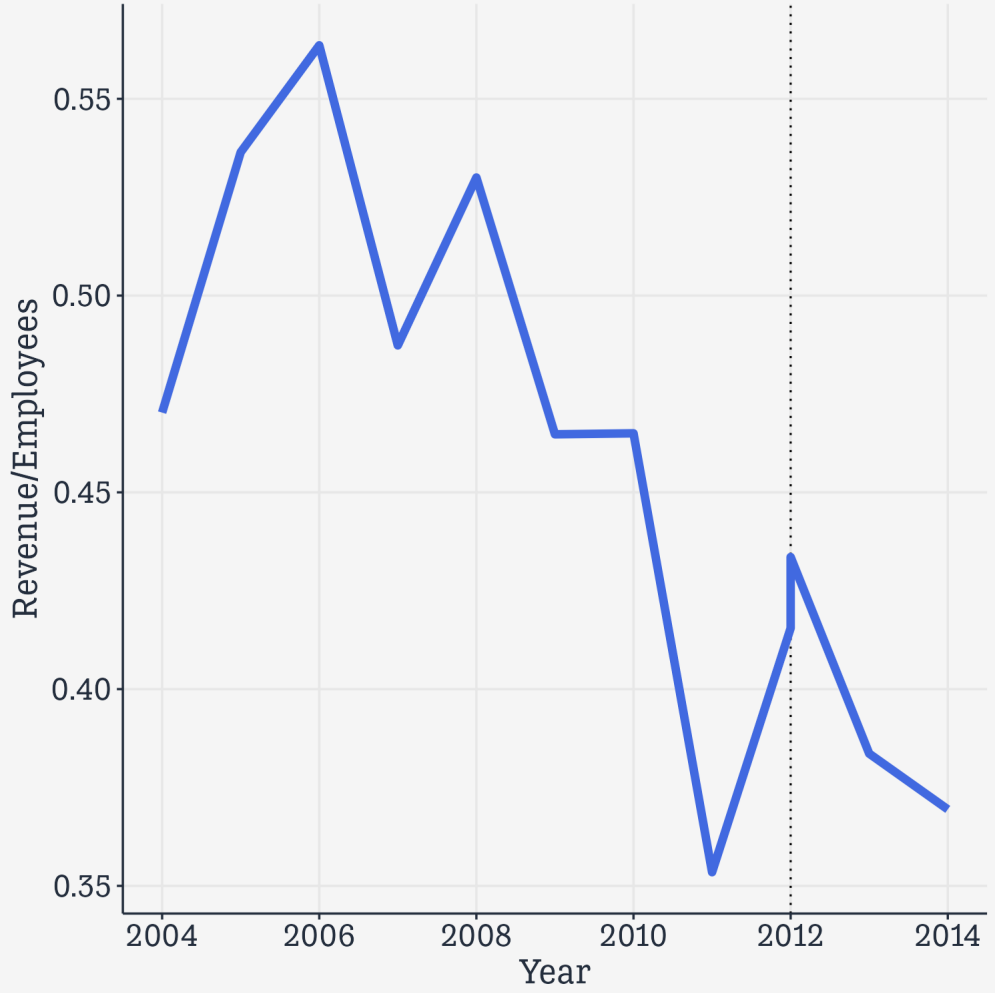
Alternatively ...

```
yahoo ▷
  ggplot(mapping =
           aes(x = Year,
               y = Revenue/Employees)) +
  geom_vline(xintercept = 2012,
             linewidth = rel(0.5),
             linetype = "dotted")
```



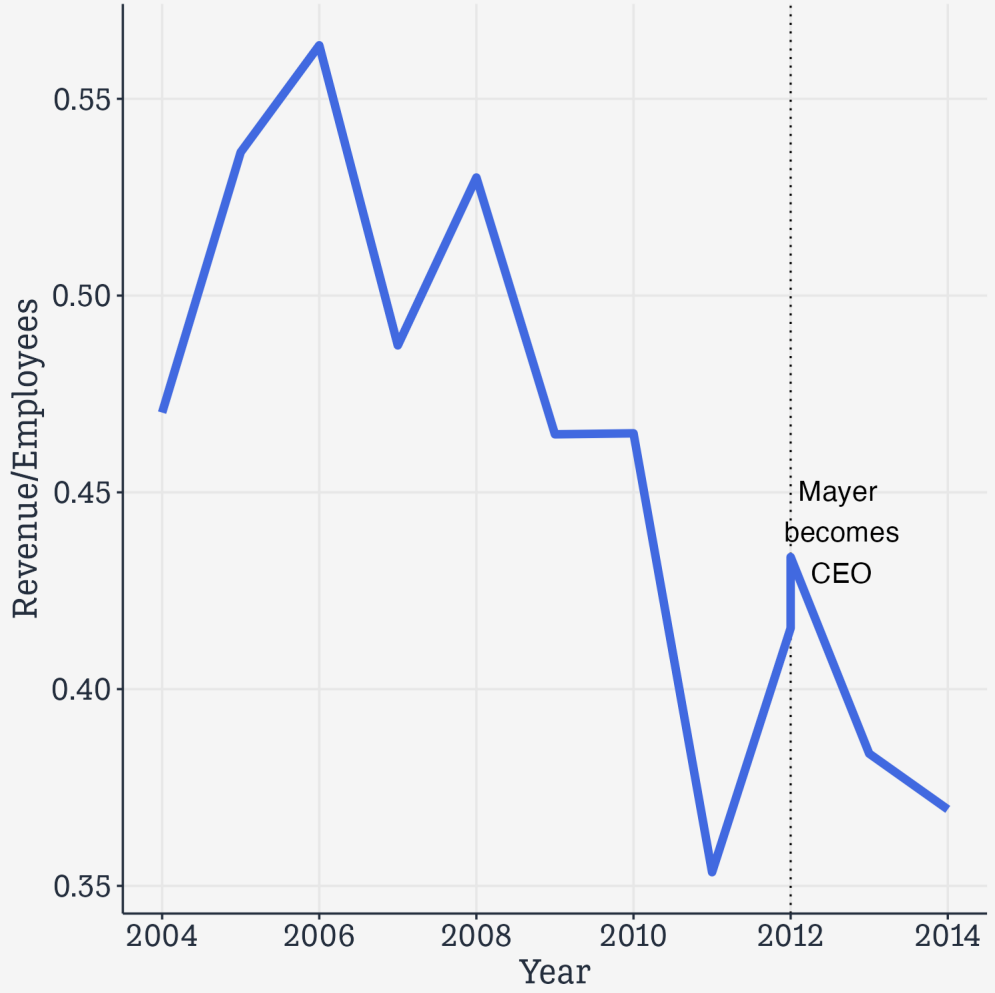
Alternatively ...

```
yahoo ▷
  ggplot(mapping =
           aes(x = Year,
               y = Revenue/Employees)) +
  geom_vline(xintercept = 2012,
             linewidth = rel(0.5),
             linetype = "dotted") +
  geom_line(color = "royalblue", linewidth
```



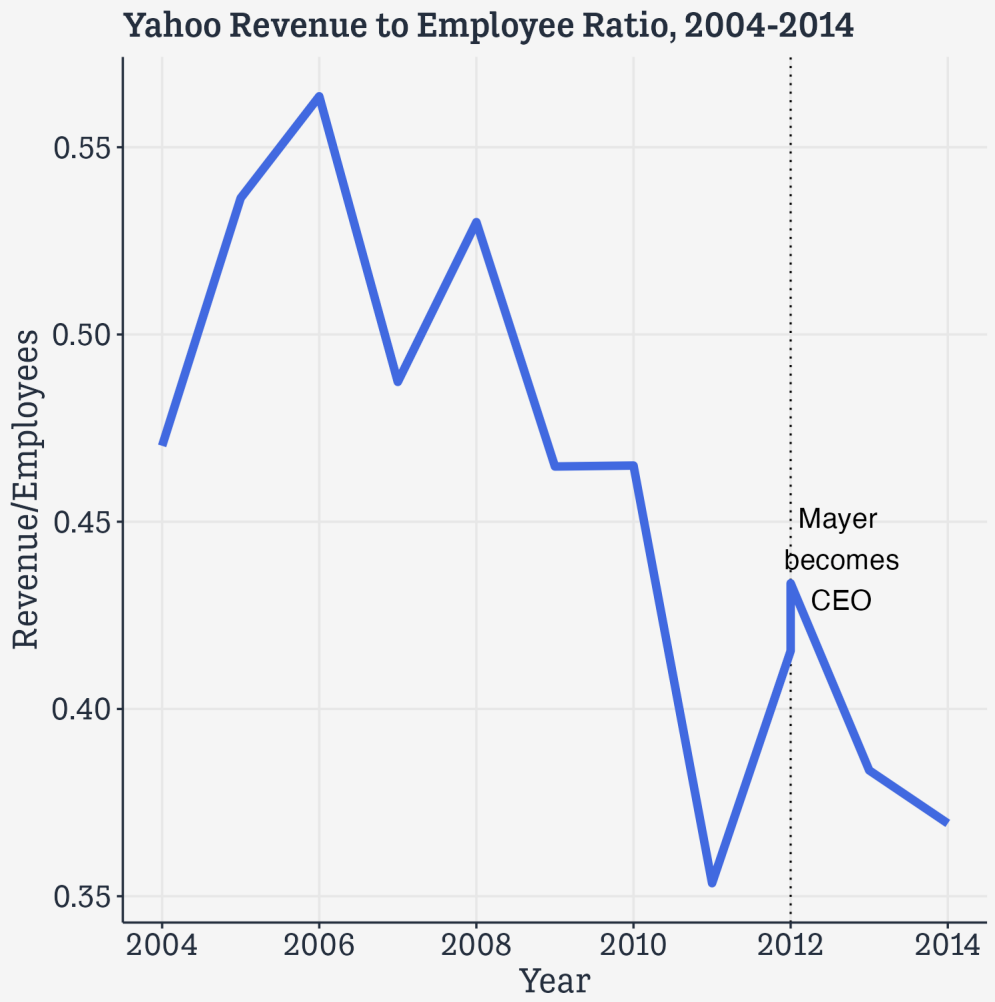
Alternatively ...

```
yahoo ▷
  ggplot(mapping =
           aes(x = Year,
               y = Revenue/Employees)) +
  geom_vline(xintercept = 2012,
             linewidth = rel(0.5),
             linetype = "dotted") +
  geom_line(color = "royalblue", linewidth
  annotate("text", x = 2012.6, y = 0.44,
           label = "Mayer\n becomes\n CEO",
```



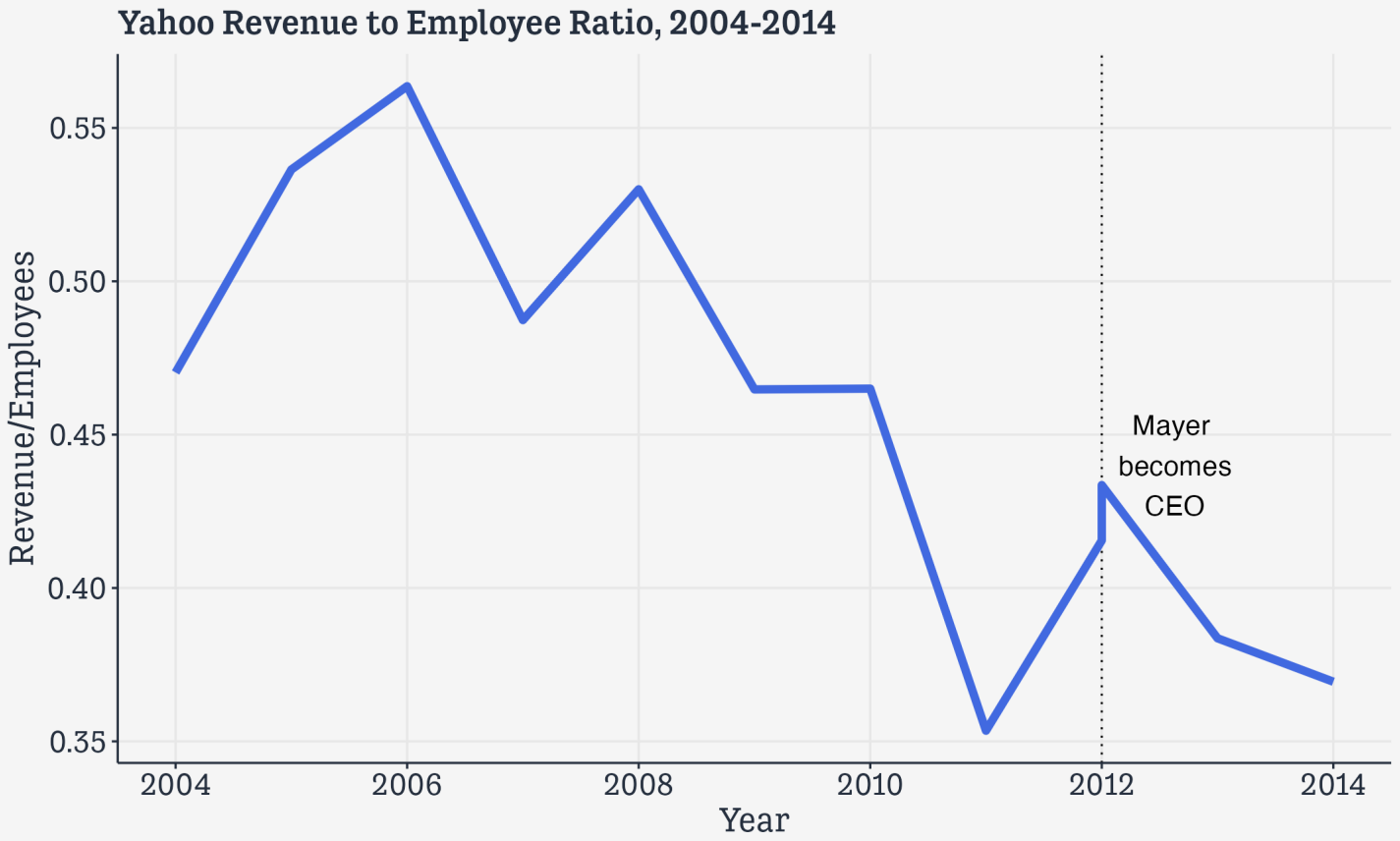
Alternatively ...

```
yahoo ▷
  ggplot(mapping =
           aes(x = Year,
               y = Revenue/Employees)) +
  geom_vline(xintercept = 2012,
             linewidth = rel(0.5),
             linetype = "dotted") +
  geom_line(color = "royalblue", linewidth
  annotate("text", x = 2012.6, y = 0.44,
           label = "Mayer\n becomes\n CEO",
  labs(title = "Yahoo Revenue to Employee R
       x = "Year",
       y = "Revenue/Employees")
```



Alternatively ...

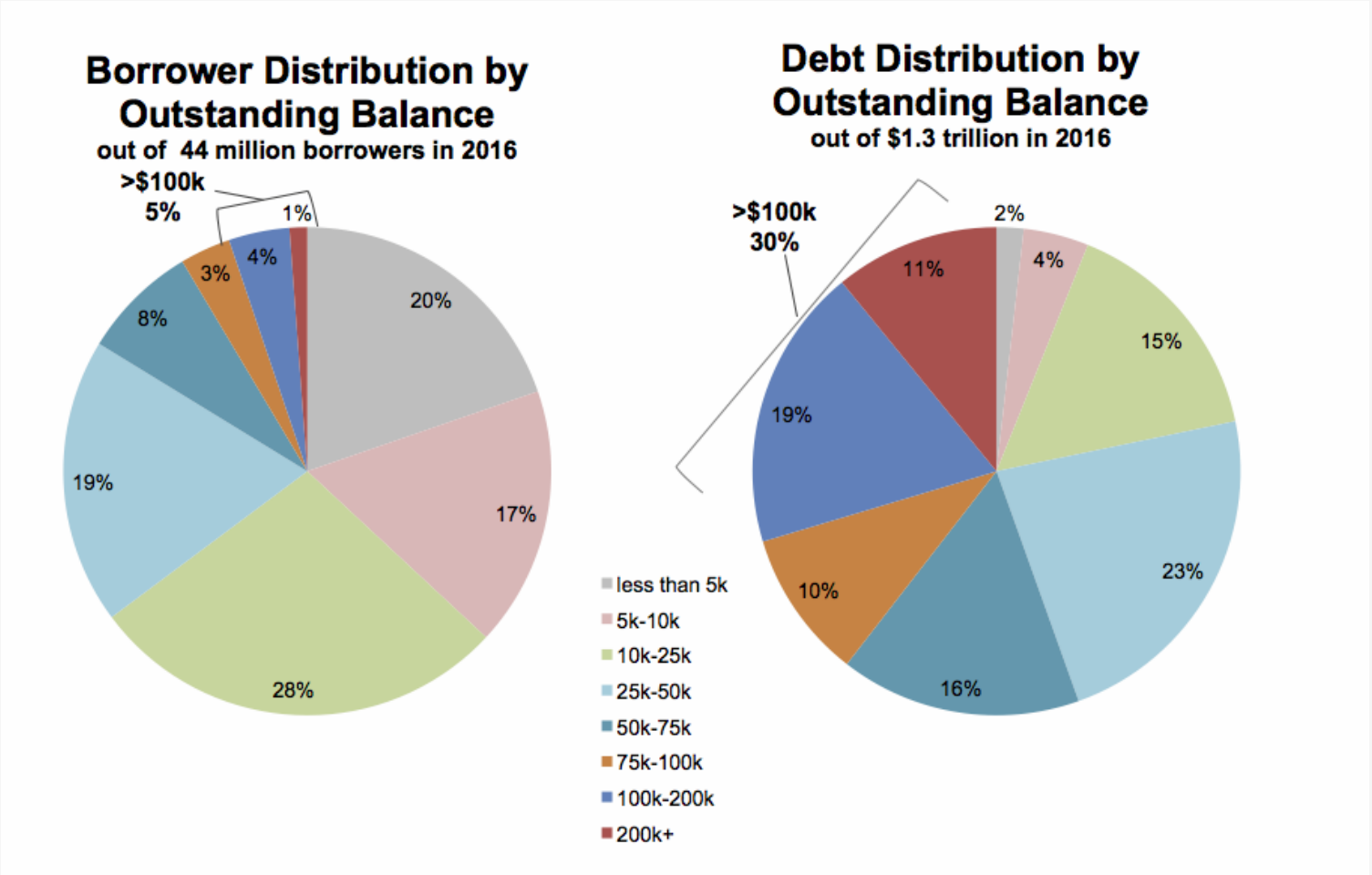
```
yahoo ▷
  ggplot(mapping =
           aes(x = Year,
               y = Revenue/Employees)) +
  geom_vline(xintercept = 2012,
             linewidth = rel(0.5),
             linetype = "dotted") +
  geom_line(color = "royalblue", linewidth
  annotate("text", x = 2012.6, y = 0.44,
           label = "Mayer\n becomes\n CEO",
  labs(title = "Yahoo Revenue to Employee R
       x = "Year",
       y = "Revenue/Employees") ⟶
  yahoo2
```



If you're interested in the ratio, just show the ratio.

Say no to pie

Pie charts are easy to mess up



Federal Reserve Bank of New York.

The data

```
studebt
```

```
## # A tibble: 16 × 4
##    Debt      type        pct Debtrc
##    <ord>     <fct>     <int> <ord>
##  1 Under $5  Borrowers    20 Under $5
##  2 $5-$10    Borrowers    17 $5-$10
##  3 $10-$25   Borrowers    28 $10-$25
##  4 $25-$50   Borrowers    19 $25-$50
##  5 $50-$75   Borrowers     8 $50-$75
##  6 $75-$100  Borrowers     3 $75-$100
##  7 $100-$200 Borrowers     4 $100-$200
##  8 Over $200 Borrowers     1 Over $200
##  9 Under $5  Balances      2 Under $5
## 10 $5-$10    Balances      4 $5-$10
## 11 $10-$25   Balances     15 $10-$25
## 12 $25-$50   Balances     23 $25-$50
## 13 $50-$75   Balances     16 $50-$75
## 14 $75-$100  Balances     10 $75-$100
## 15 $100-$200 Balances     19 $100-$200
## 16 Over $200 Balances     11 Over $200
```

`Debt` and `Debtrc` are both ordered factors.

A little prep work

```
p_ylab ← "Amount Owed, in thousands of Dollars"
p_title ← "Outstanding Student Loans"
p_subtitle ← "44 million borrowers owe a total of $1.3 trillion"
p_caption ← "Source: FRB NY"

studebt ← studebt ▷
  mutate(type_label = recode(type, "Borrowers" = "Percent of all Borrowers",
                         "Balances" = "Percent of all Balances"))

studebt
```

```
## # A tibble: 16 × 5
##    Debt      type        pct Debtrc    type_label
##    <ord>     <fct>     <int> <ord>     <fct>
##  1 Under $5  Borrowers    20 Under $5  Percent of all Borrowers
##  2 $5-$10    Borrowers    17 $5-$10    Percent of all Borrowers
##  3 $10-$25   Borrowers    28 $10-$25   Percent of all Borrowers
##  4 $25-$50   Borrowers    19 $25-$50   Percent of all Borrowers
##  5 $50-$75   Borrowers     8 $50-$75   Percent of all Borrowers
##  6 $75-$100  Borrowers     3 $75-$100  Percent of all Borrowers
##  7 $100-$200 Borrowers     4 $100-$200 Percent of all Borrowers
##  8 Over $200 Borrowers     1 Over $200 Percent of all Borrowers
##  9 Under $5  Balances      2 Under $5  Percent of all Balances
## 10 $5-$10    Balances      4 $5-$10    Percent of all Balances
## 11 $10-$25   Balances     15 $10-$25   Percent of all Balances
## 12 $25-$50   Balances     23 $25-$50   Percent of all Balances
## 13 $50-$75   Balances     16 $50-$75   Percent of all Balances
## 14 $75-$100  Balances     10 $75-$100  Percent of all Balances
## 15 $100-$200 Balances     19 $100-$200 Percent of all Balances
## 16 Over $200 Balances     11 Over $200 Percent of all Balances
```

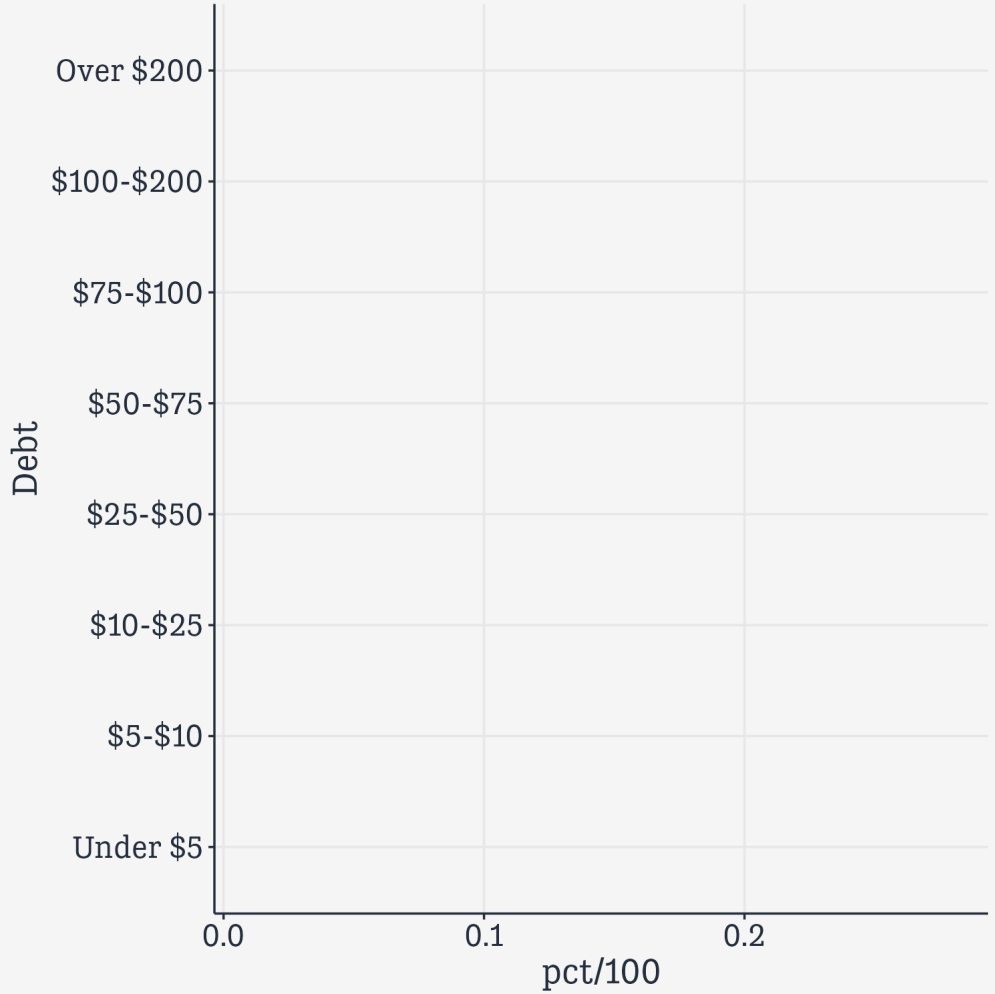
Debt Plot 1

```
studebt
```

```
## # A tibble: 16 × 5
##    Debt      type        pct Debtrc    type_label
##    <ord>     <fct>     <int> <ord>     <fct>
##  1 Under $5  Borrowers    20 Under $5  Percent of all Borrowers
##  2 $5-$10    Borrowers    17 $5-$10    Percent of all Borrowers
##  3 $10-$25   Borrowers    28 $10-$25   Percent of all Borrowers
##  4 $25-$50   Borrowers    19 $25-$50   Percent of all Borrowers
##  5 $50-$75   Borrowers     8 $50-$75   Percent of all Borrowers
##  6 $75-$100  Borrowers     3 $75-$100  Percent of all Borrowers
##  7 $100-$200 Borrowers     4 $100-$200 Percent of all Borrowers
##  8 Over $200 Borrowers     1 Over $200 Percent of all Borrowers
##  9 Under $5  Balances      2 Under $5  Percent of all Balances
## 10 $5-$10    Balances      4 $5-$10    Percent of all Balances
## 11 $10-$25   Balances     15 $10-$25   Percent of all Balances
## 12 $25-$50   Balances     23 $25-$50   Percent of all Balances
## 13 $50-$75   Balances     16 $50-$75   Percent of all Balances
## 14 $75-$100  Balances     10 $75-$100  Percent of all Balances
## 15 $100-$200 Balances     19 $100-$200 Percent of all Balances
## 16 Over $200 Balances     11 Over $200 Percent of all Balances
```

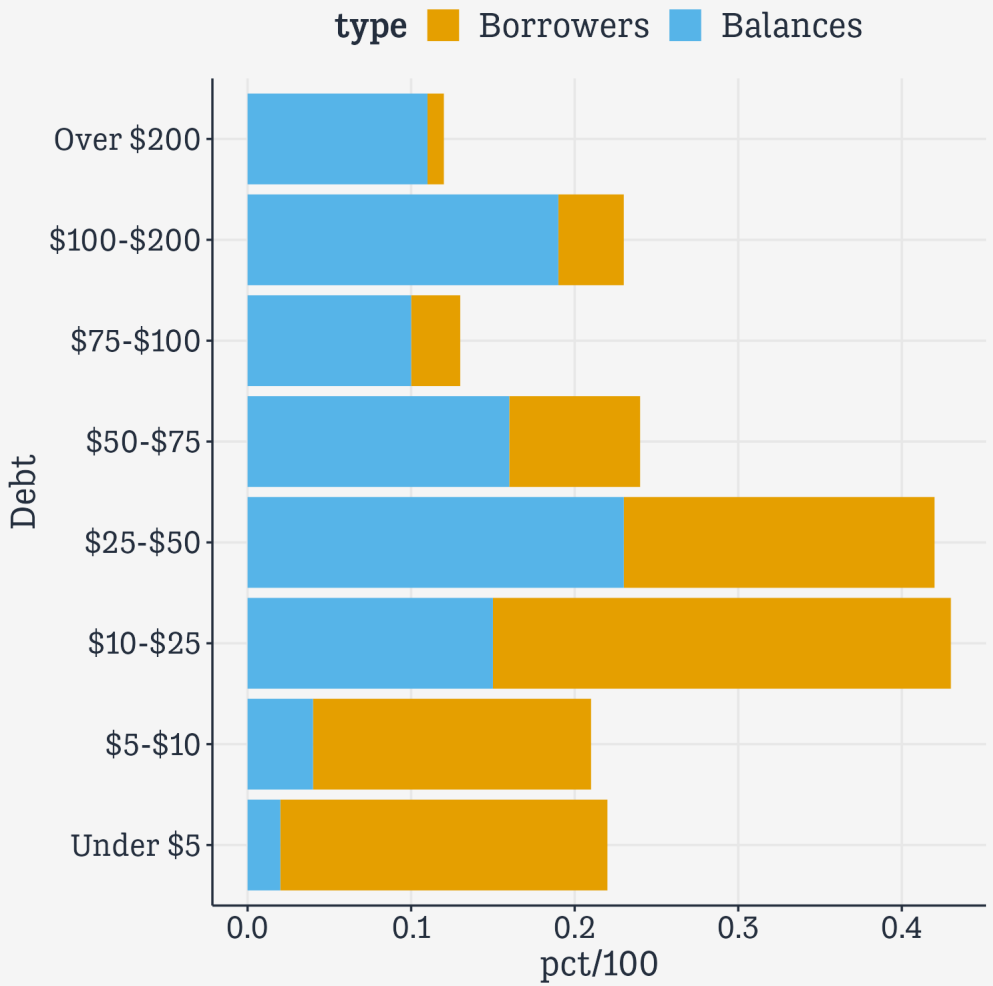
Debt Plot 1

```
studebt ▷
  ggplot(mapping =
           aes(x = pct/100,
               y = Debt,
               fill = type))
```



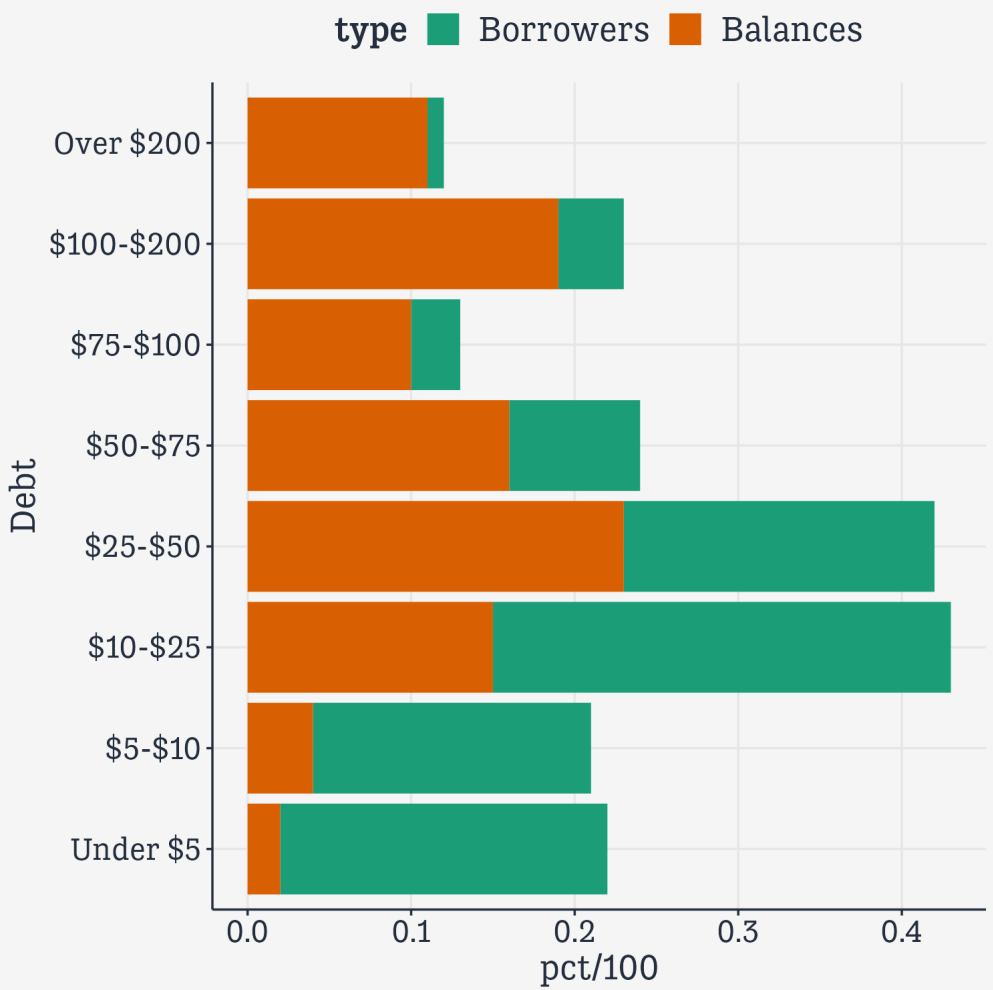
Debt Plot 1

```
studebt ▷
  ggplot(mapping =
           aes(x = pct/100,
               y = Debt,
               fill = type)) +
  geom_col()
```



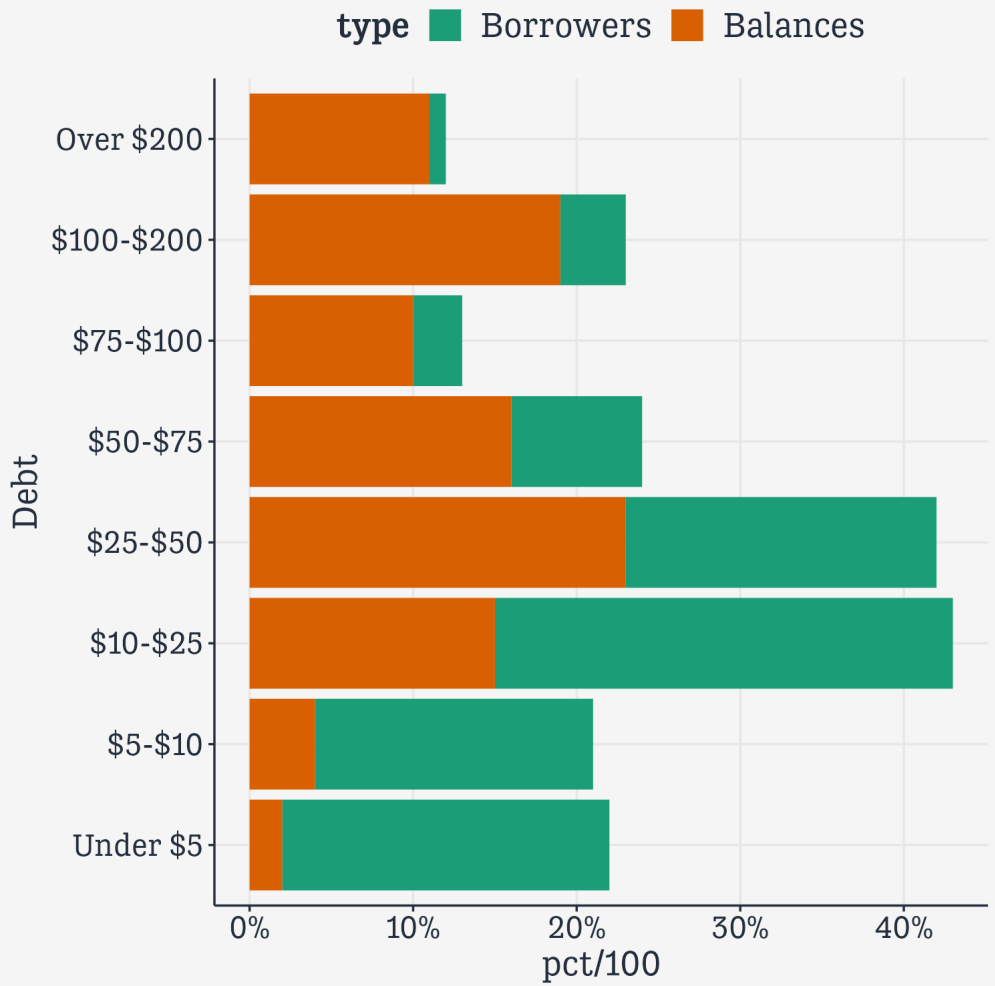
Debt Plot 1

```
studebt ▷
  ggplot(mapping =
           aes(x = pct/100,
               y = Debt,
               fill = type)) +
  geom_col() +
  scale_fill_brewer(type = "qual",
                    palette = "Dark2")
```



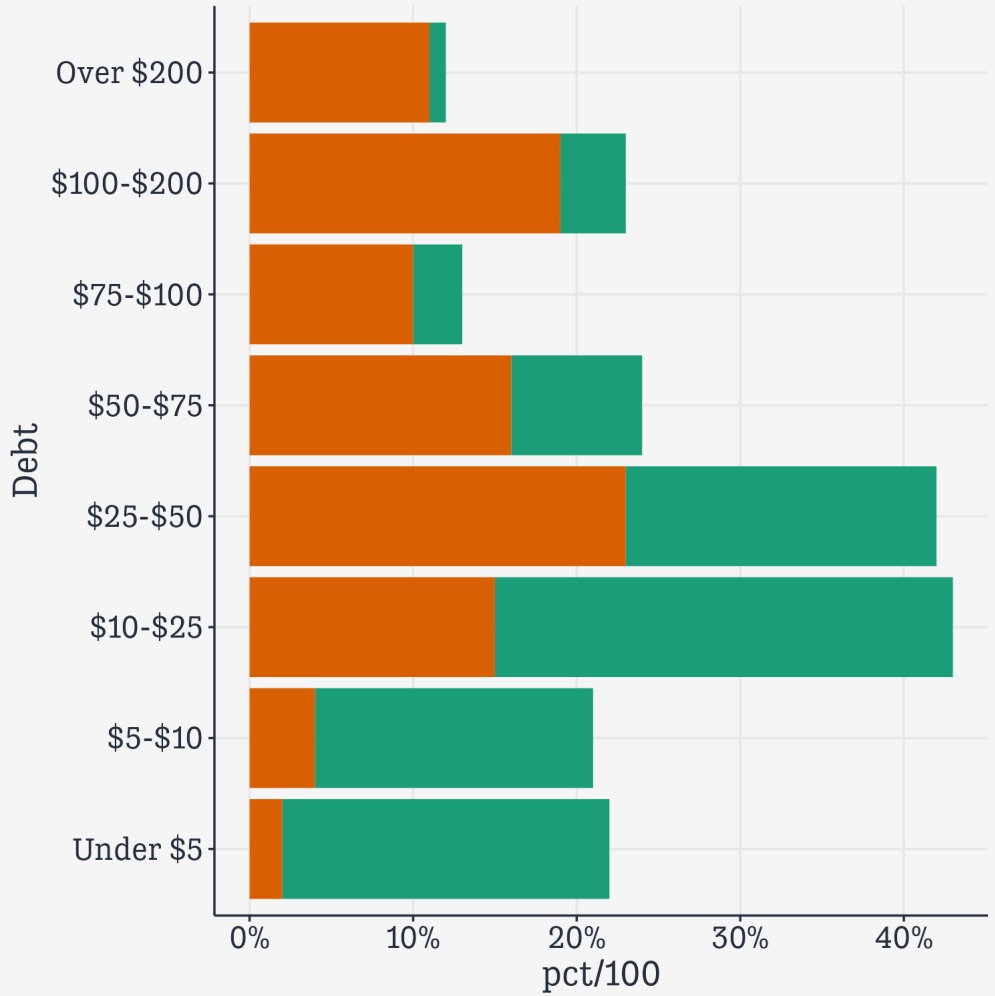
Debt Plot 1

```
studebt ▷
  ggplot(mapping =
           aes(x = pct/100,
               y = Debt,
               fill = type)) +
  geom_col() +
  scale_fill_brewer(type = "qual",
                    palette = "Dark2") +
  scale_x_continuous(labels = label_percent())
```



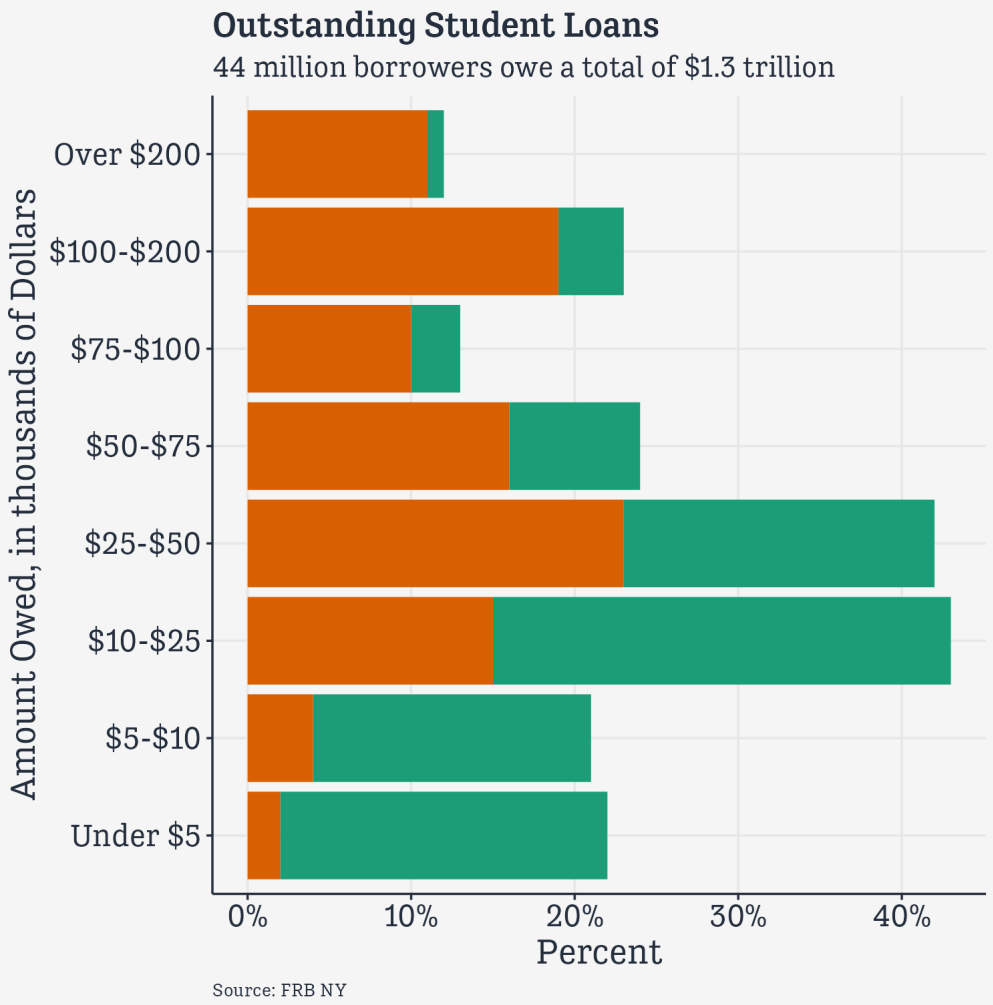
Debt Plot 1

```
studebt ▷
  ggplot(mapping =
           aes(x = pct/100,
               y = Debt,
               fill = type)) +
  geom_col() +
  scale_fill_brewer(type = "qual",
                    palette = "Dark2") +
  scale_x_continuous(labels = label_percent()) +
  guides(fill = "none")
```



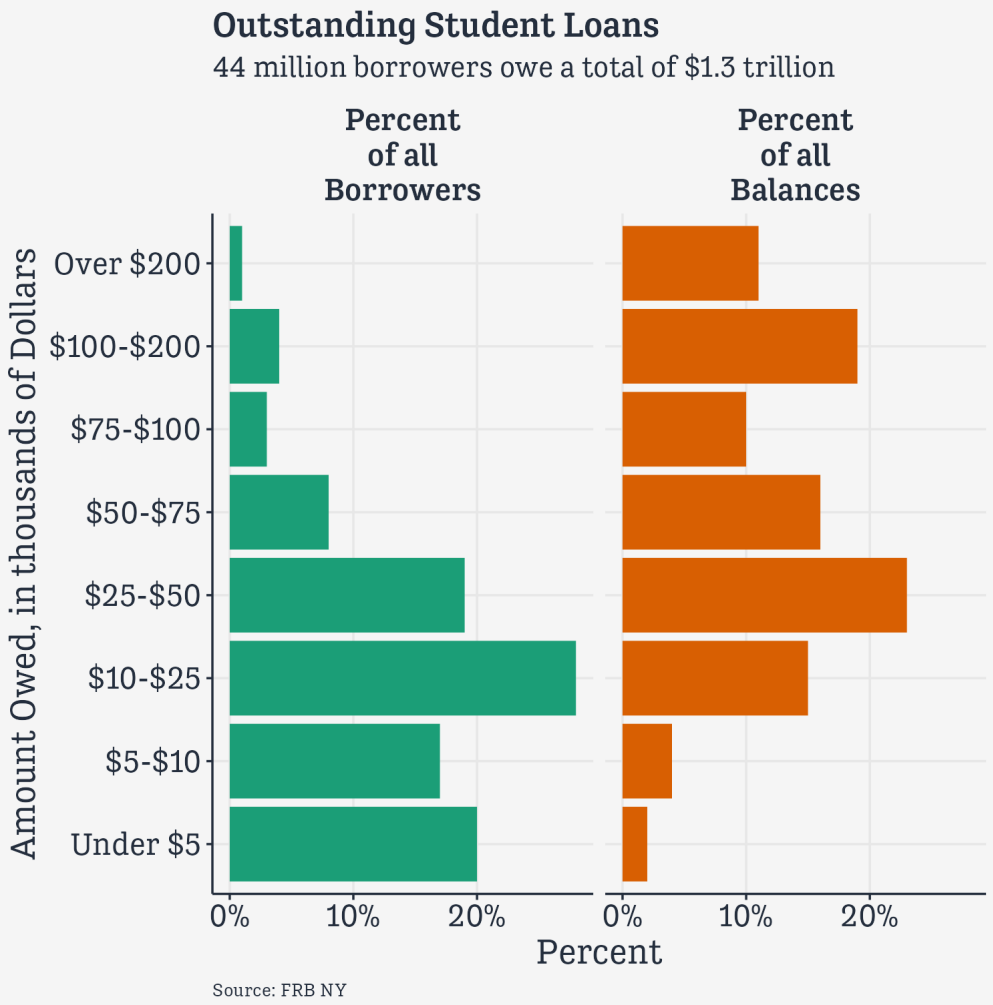
Debt Plot 1

```
studebt ▷
  ggplot(mapping =
           aes(x = pct/100,
               y = Debt,
               fill = type)) +
  geom_col() +
  scale_fill_brewer(type = "qual",
                    palette = "Dark2") +
  scale_x_continuous(labels = label_percent()) +
  guides(fill = "none") +
  labs(x = "Percent",
       y = p_ylab,
       caption = p_caption,
       title = p_title,
       subtitle = p_subtitle)
```



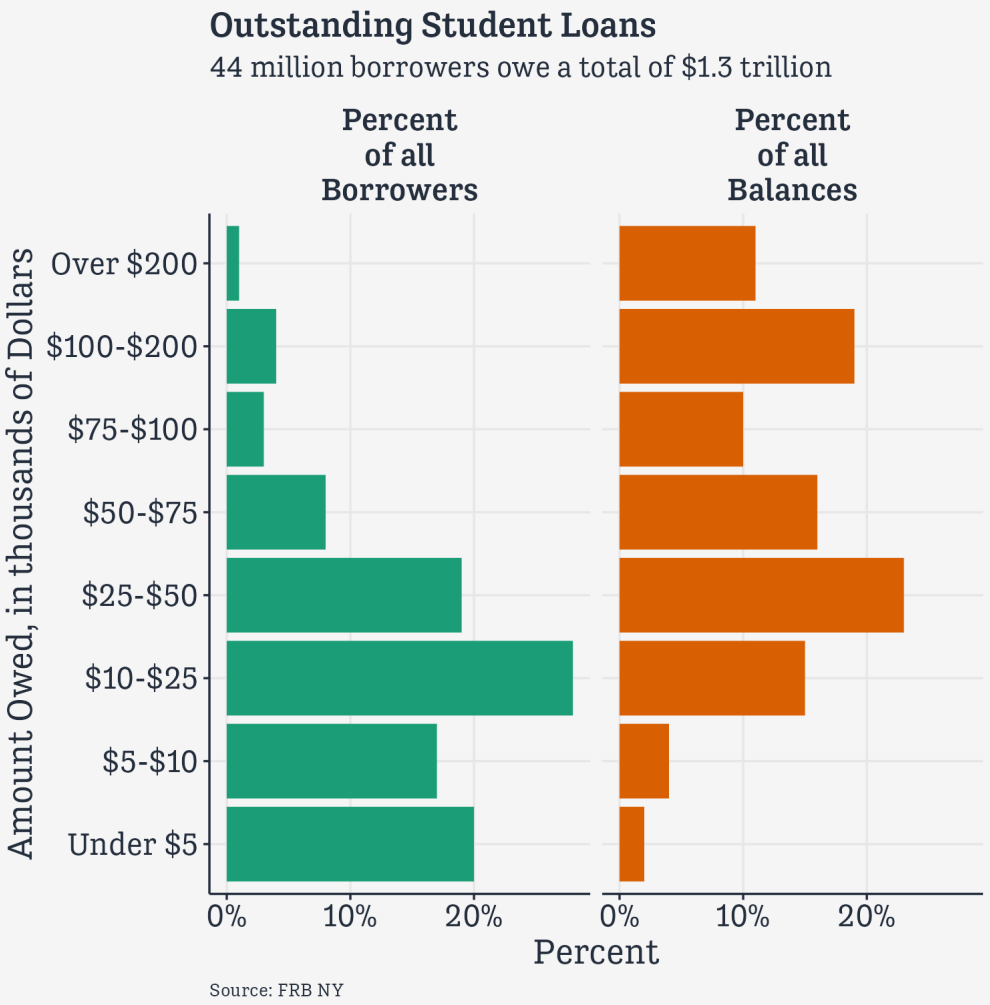
Debt Plot 1

```
studebt ▷
  ggplot(mapping =
           aes(x = pct/100,
               y = Debt,
               fill = type)) +
  geom_col() +
  scale_fill_brewer(type = "qual",
                    palette = "Dark2") +
  scale_x_continuous(labels = label_percent()) +
  guides(fill = "none") +
  labels(x = "Percent",
       y = p_ylab,
       caption = p_caption,
       title = p_title,
       subtitle = p_subtitle) +
  facet_wrap(~ type_label,
             labeller =
               label_wrap_gen(width=10))
```



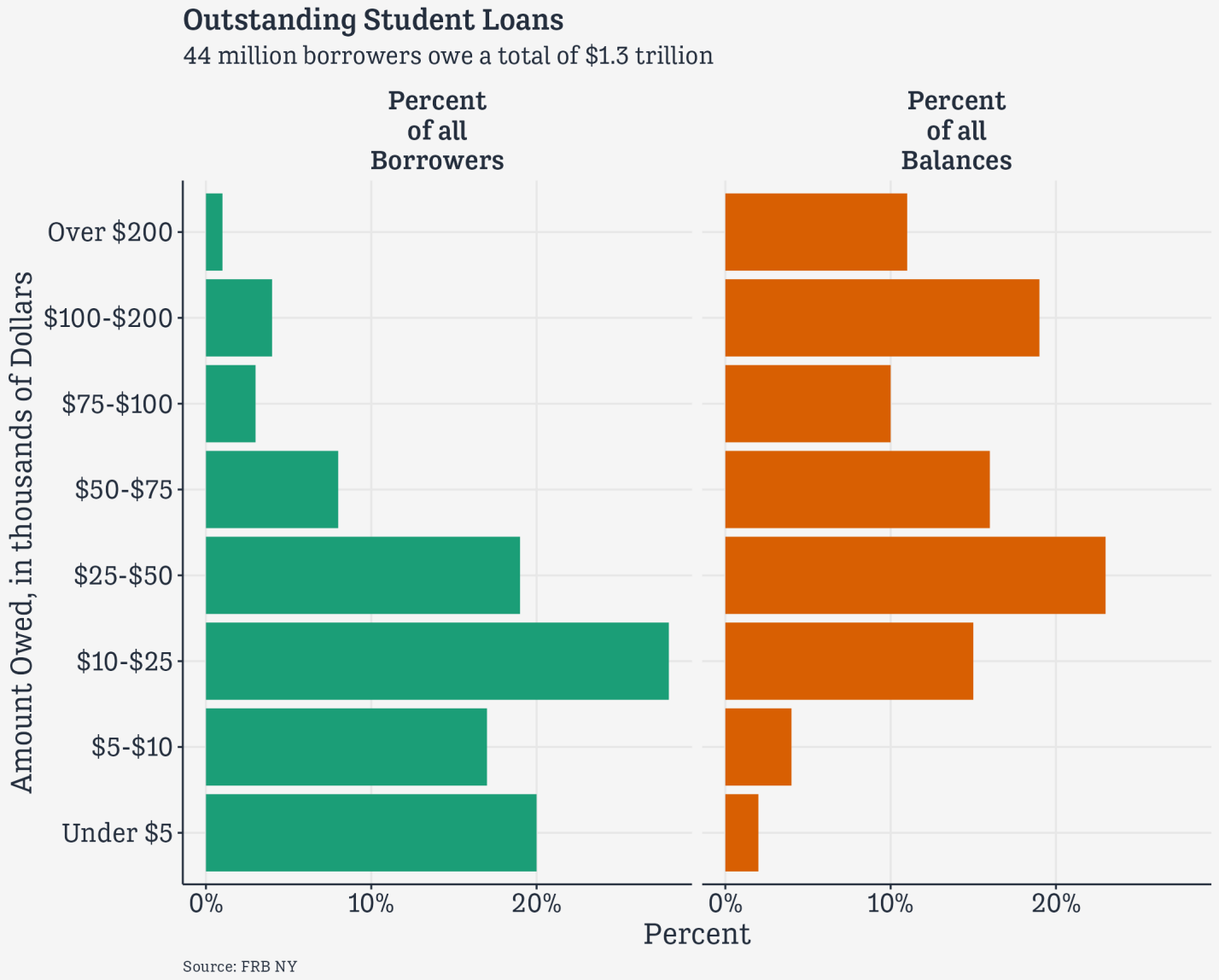
Debt Plot 1

```
studebt ▷
  ggplot(mapping =
           aes(x = pct/100,
               y = Debt,
               fill = type)) +
  geom_col() +
  scale_fill_brewer(type = "qual",
                    palette = "Dark2") +
  scale_x_continuous(labels = label_percent()) +
  guides(fill = "none") +
  labs(x = "Percent",
       y = p_ylab,
       caption = p_caption,
       title = p_title,
       subtitle = p_subtitle) +
  facet_wrap(~ type_label,
             labeller =
               label_wrap_gen(width=10)) +
    theme(strip.text.x =
          element_text(face = "bold"))
```



Debt Plot 1

```
studebt ▷
  ggplot(mapping =
           aes(x = pct/100,
               y = Debt,
               fill = type)) +
  geom_col() +
  scale_fill_brewer(type = "qual",
                    palette = "Dark2") +
  scale_x_continuous(labels = label_percent()) +
  guides(fill = "none") +
  labs(x = "Percent",
       y = p_ylab,
       caption = p_caption,
       title = p_title,
       subtitle = p_subtitle) +
  facet_wrap(~ type_label,
             labeller =
               label_wrap_gen(width=10)) +
    theme(strip.text.x =
          element_text(face = "bold")) →
  p1_debt
```



Pies redrawn as facets

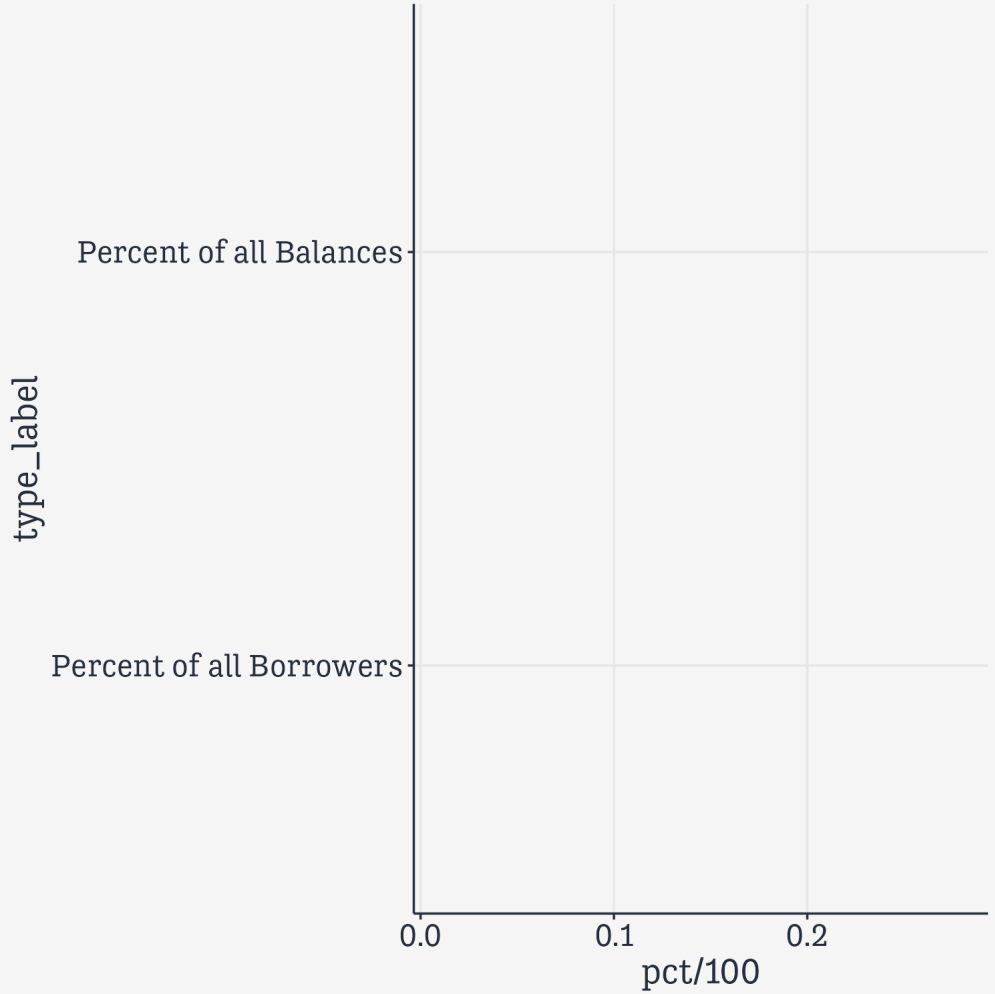
Alternatively, as a kind of stacked bar chart

```
studebt
```

```
## # A tibble: 16 × 5
##    Debt      type        pct Debtrc    type_label
##    <ord>     <fct>     <int> <ord>     <fct>
##  1 Under $5  Borrowers    20 Under $5  Percent of all Borrowers
##  2 $5-$10    Borrowers    17 $5-$10    Percent of all Borrowers
##  3 $10-$25   Borrowers    28 $10-$25   Percent of all Borrowers
##  4 $25-$50   Borrowers    19 $25-$50   Percent of all Borrowers
##  5 $50-$75   Borrowers     8 $50-$75   Percent of all Borrowers
##  6 $75-$100  Borrowers     3 $75-$100  Percent of all Borrowers
##  7 $100-$200 Borrowers     4 $100-$200 Percent of all Borrowers
##  8 Over $200 Borrowers     1 Over $200 Percent of all Borrowers
##  9 Under $5  Balances      2 Under $5  Percent of all Balances
## 10 $5-$10    Balances      4 $5-$10    Percent of all Balances
## 11 $10-$25   Balances     15 $10-$25   Percent of all Balances
## 12 $25-$50   Balances     23 $25-$50   Percent of all Balances
## 13 $50-$75   Balances     16 $50-$75   Percent of all Balances
## 14 $75-$100  Balances     10 $75-$100  Percent of all Balances
## 15 $100-$200 Balances     19 $100-$200 Percent of all Balances
## 16 Over $200 Balances     11 Over $200 Percent of all Balances
```

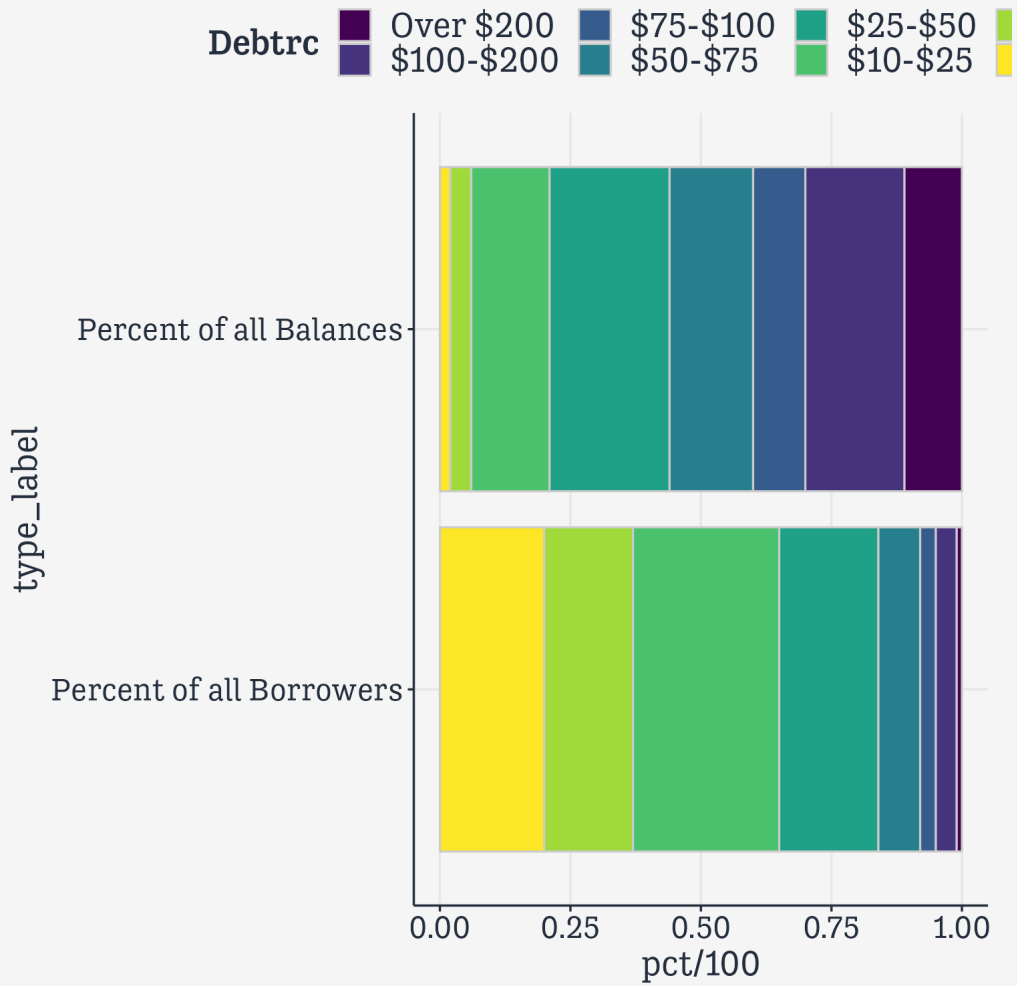
Alternatively, as a kind of stacked bar chart

```
studebt ▷
  ggplot(mapping = aes(x = pct/100,
                       y = type_label,
                       fill = Debtrc))
```



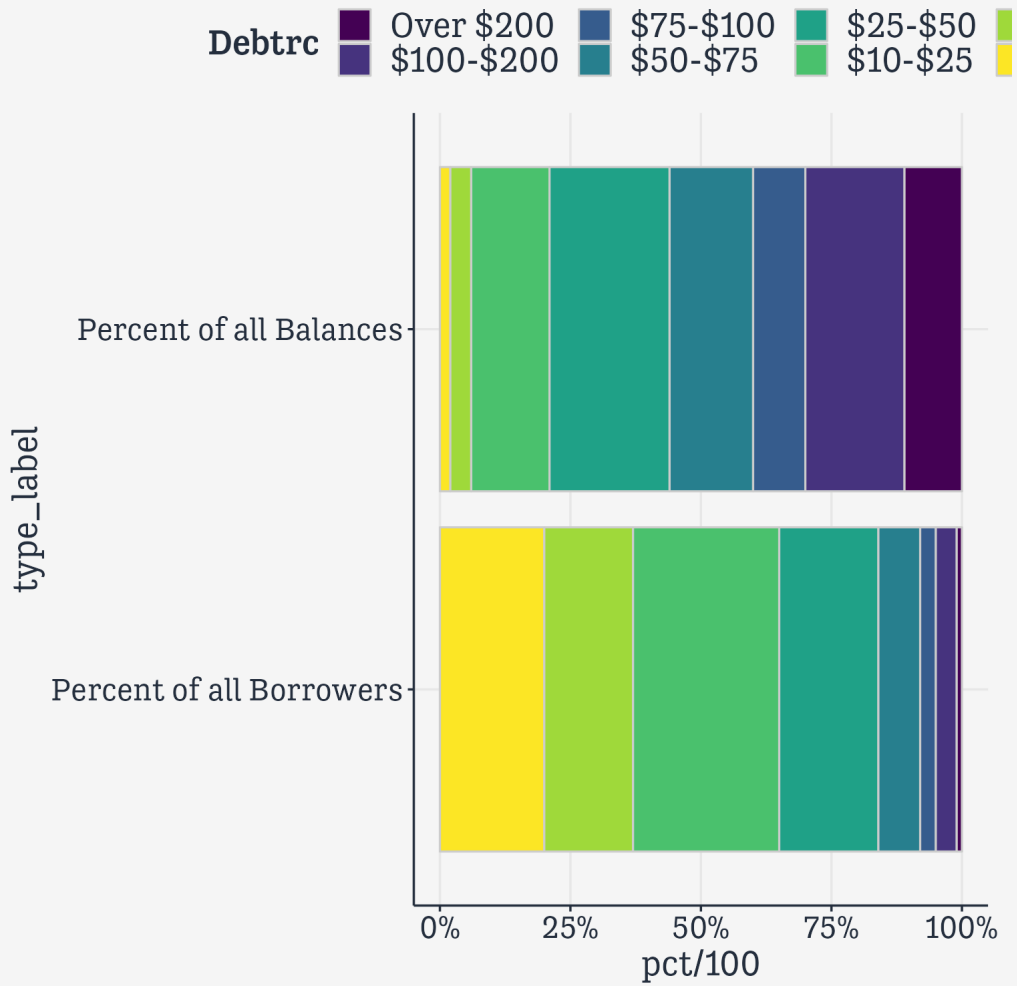
Alternatively, as a kind of stacked bar chart

```
studebt ▷
  ggplot(mapping = aes(x = pct/100,
                       y = type_label,
                       fill = Debtrc)) +
  geom_col(color = "gray80")
```



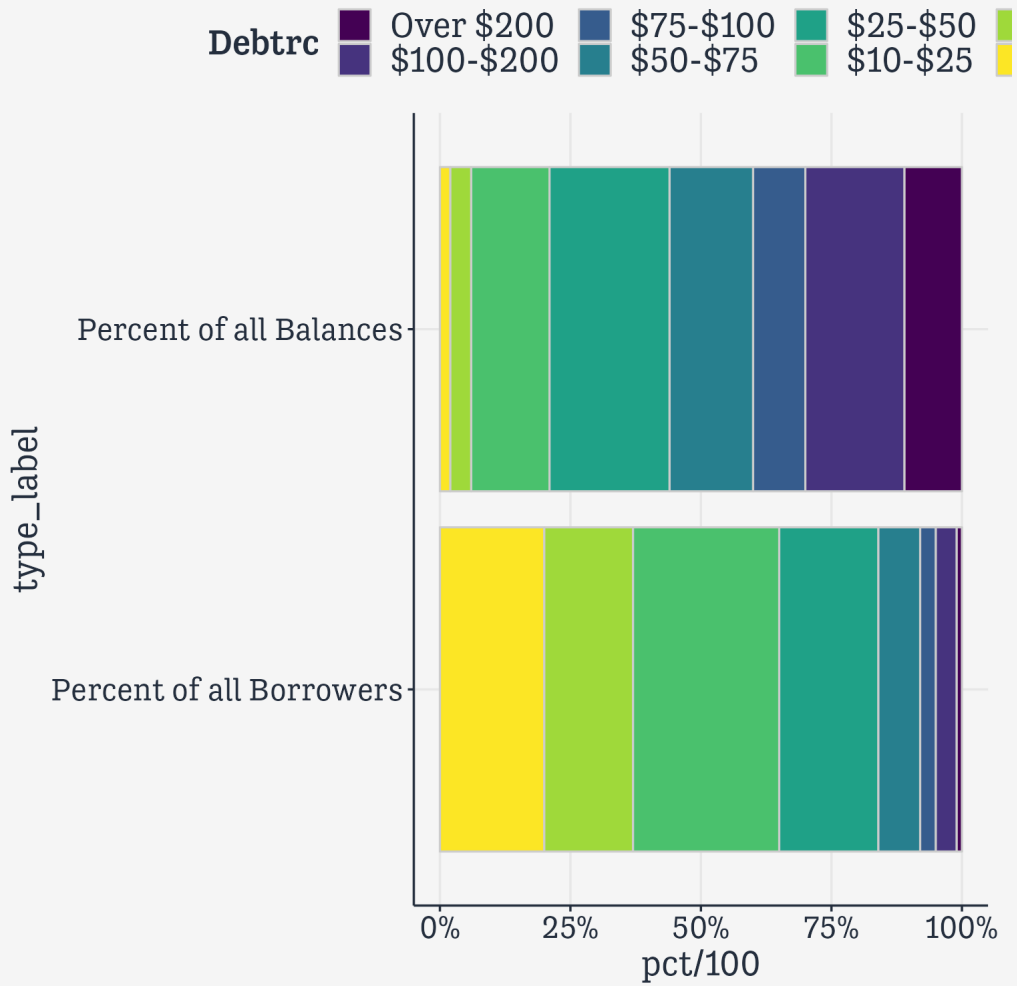
Alternatively, as a kind of stacked bar chart

```
studebt ▷
  ggplot(mapping = aes(x = pct/100,
                       y = type_label,
                       fill = Debtrc)) +
  geom_col(color = "gray80") +
  scale_x_continuous(labels =
                       label_percent())
```



Alternatively, as a kind of stacked bar chart

```
studebt ▷
  ggplot(mapping = aes(x = pct/100,
                       y = type_label,
                       fill = Debtrc)) +
  geom_col(color = "gray80") +
  scale_x_continuous(labels =
                       label_percent()) +
  scale_fill_viridis_d()
```

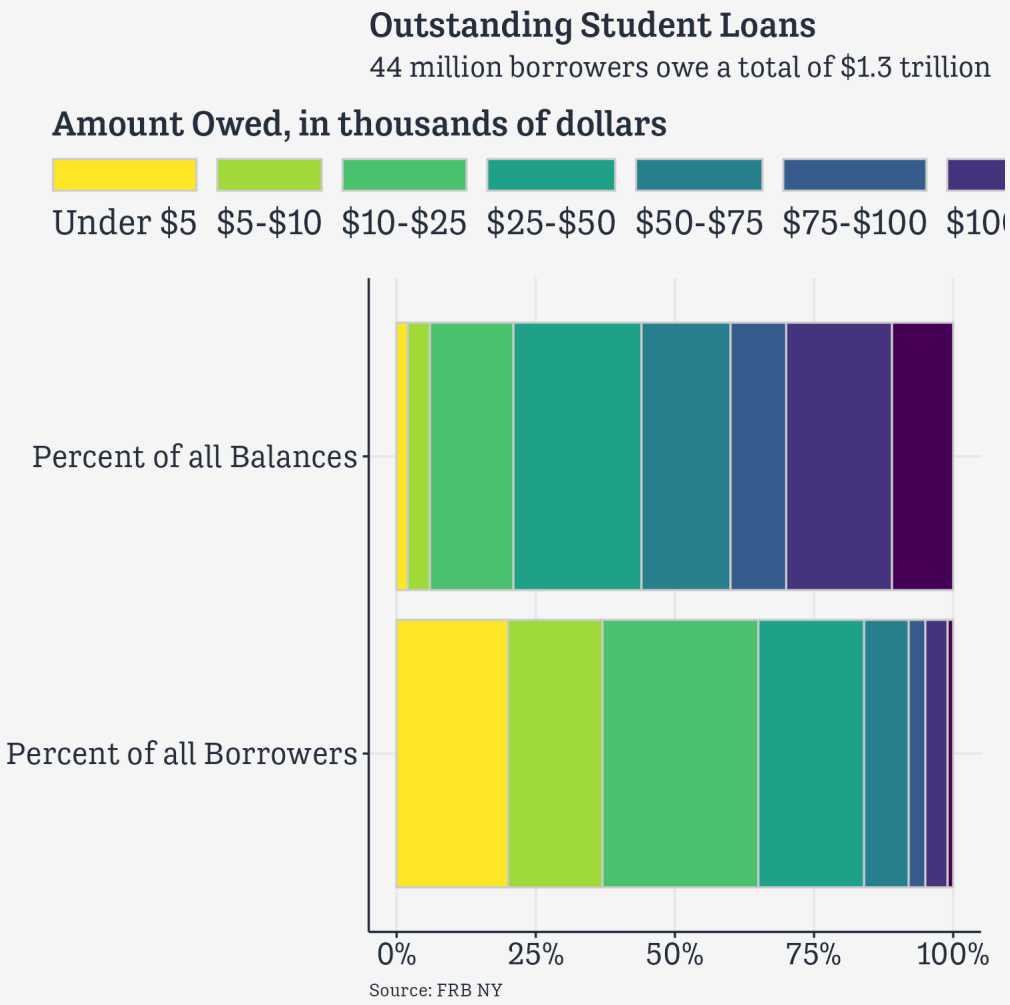


Alternatively, as a kind of stacked bar chart

```
studebt ▷
  ggplot(mapping = aes(x = pct/100,
                       y = type_label,
                       fill = Debtrc)) +
  geom_col(color = "gray80") +
  scale_x_continuous(labels =
                       label_percent()) +
  scale_fill_viridis_d() +
  guides(fill =
           guide_legend(reverse = TRUE,
                        title.position = "top",
                        label.position = "bottom",
                        keywidth = 3,
                        nrow = 1))
```

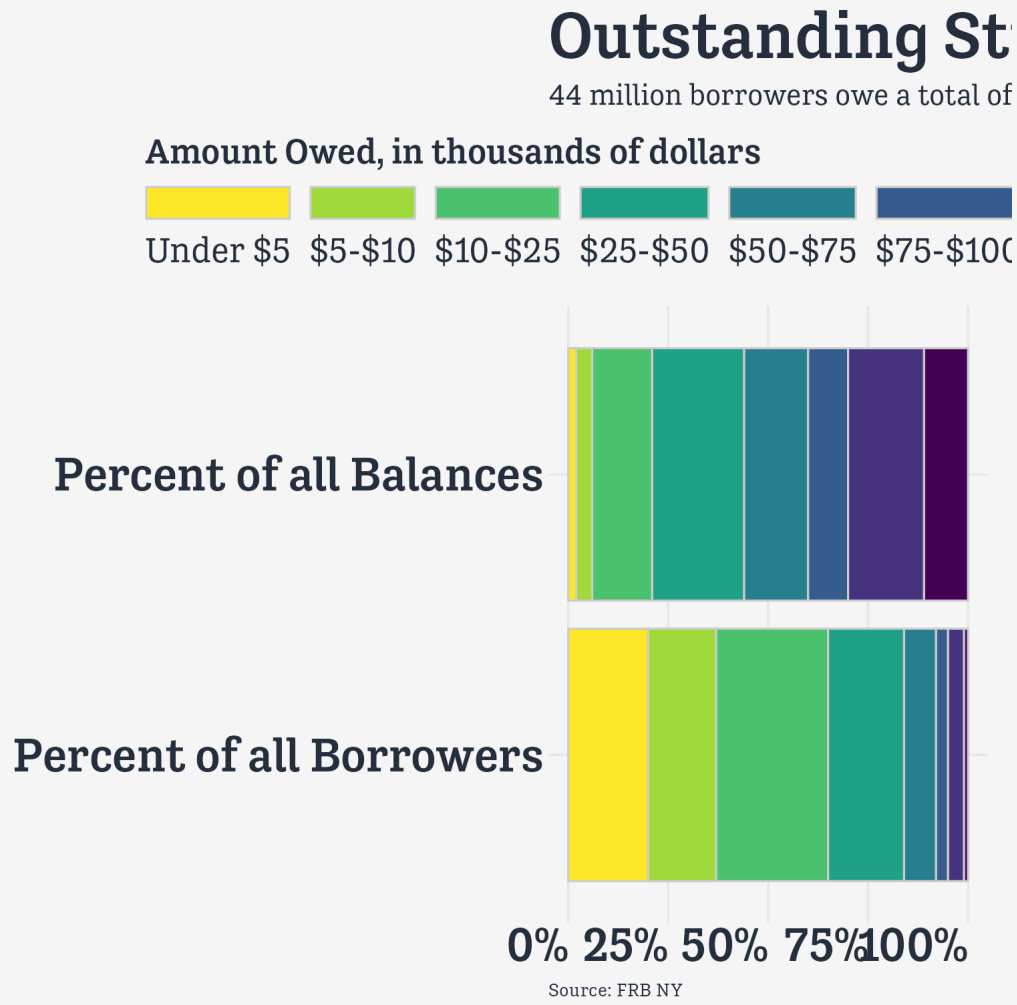
Alternatively, as a kind of stacked bar chart

```
studebt ▷
  ggplot(mapping = aes(x = pct/100,
                       y = type_label,
                       fill = Debtrc)) +
  geom_col(color = "gray80") +
  scale_x_continuous(labels =
                       label_percent()) +
  scale_fill_viridis_d() +
  guides(fill =
           guide_legend(reverse = TRUE,
                        title.position = "top",
                        label.position = "bottom",
                        keywidth = 3,
                        nrow = 1)) +
  labs(x = NULL, y = NULL,
       fill = "Amount Owed, in thousands of dollar
       caption = p_caption, title = p_title,
       subtitle = p_subtitle)
```



Alternatively, as a kind of stacked bar chart

```
studebt ▷
  ggplot(mapping = aes(x = pct/100,
                       y = type_label,
                       fill = Debtrc)) +
  geom_col(color = "gray80") +
  scale_x_continuous(labels =
                       label_percent()) +
  scale_fill_viridis_d() +
  guides(fill =
           guide_legend(reverse = TRUE,
                        title.position = "top",
                        label.position = "bottom",
                        keywidth = 3,
                        nrow = 1)) +
  labs(x = NULL, y = NULL,
       fill = "Amount Owed, in thousands of dollar
       caption = p_caption, title = p_title,
       subtitle = p_subtitle) +
  theme(legend.position = "top",
        plot.title = element_text(size = rel(2.8)
        axis.text = element_text(face = "bold",
                  hjust = 1,
                  size = rel(2)),
          axis.ticks.length = unit(0, "cm"),
          axis.line = element_blank(),
          panel.grid = element_blank())
```

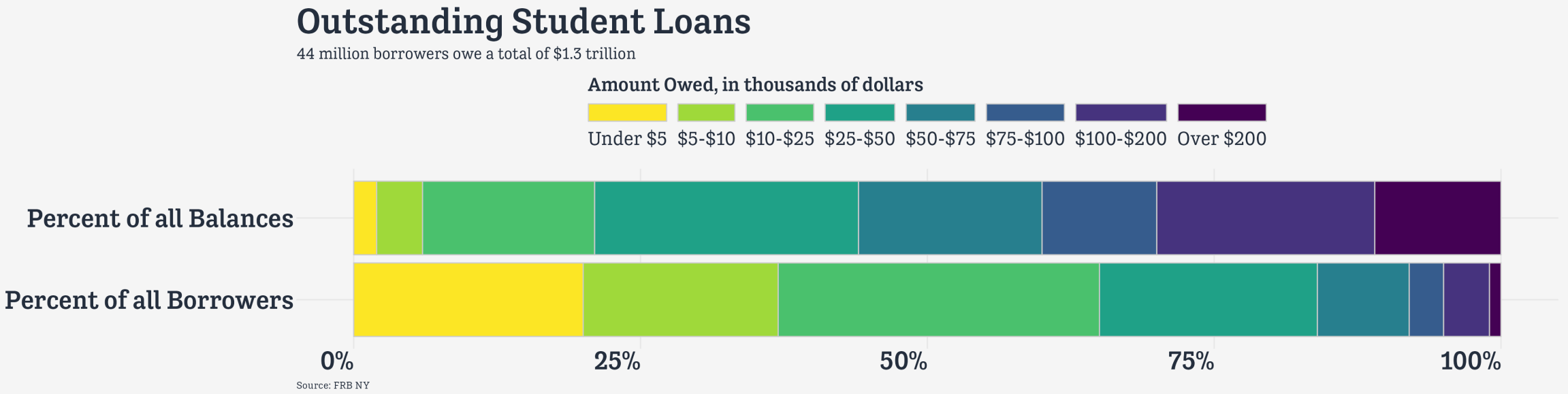


Alternatively, as a kind of stacked bar chart

```
studebt ▷
  ggplot(mapping = aes(x = pct/100,
                       y = type_label,
                       fill = Debtrc)) +
  geom_col(color = "gray80") +
  scale_x_continuous(labels =
                       label_percent()) +
  scale_fill_viridis_d() +
  guides(fill =
           guide_legend(reverse = TRUE,
                        title.position = "top",
                        label.position = "bottom",
                        keywidth = 3,
                        nrow = 1)) +
  labs(x = NULL, y = NULL,
       fill = "Amount Owed, in thousands of dollar
       caption = p_caption, title = p_title,
       subtitle = p_subtitle) +
  theme(legend.position = "top",
        plot.title = element_text(size = rel(2.8)
        axis.text = element_text(face = "bold",
                  hjust = 1,
                  size = rel(2)),
          axis.ticks.length = unit(0, "cm"),
          axis.line = element_blank(),
          panel.grid = element_blank()) -->
  p_debt2
```

Alternatively, as a kind of stacked bar chart

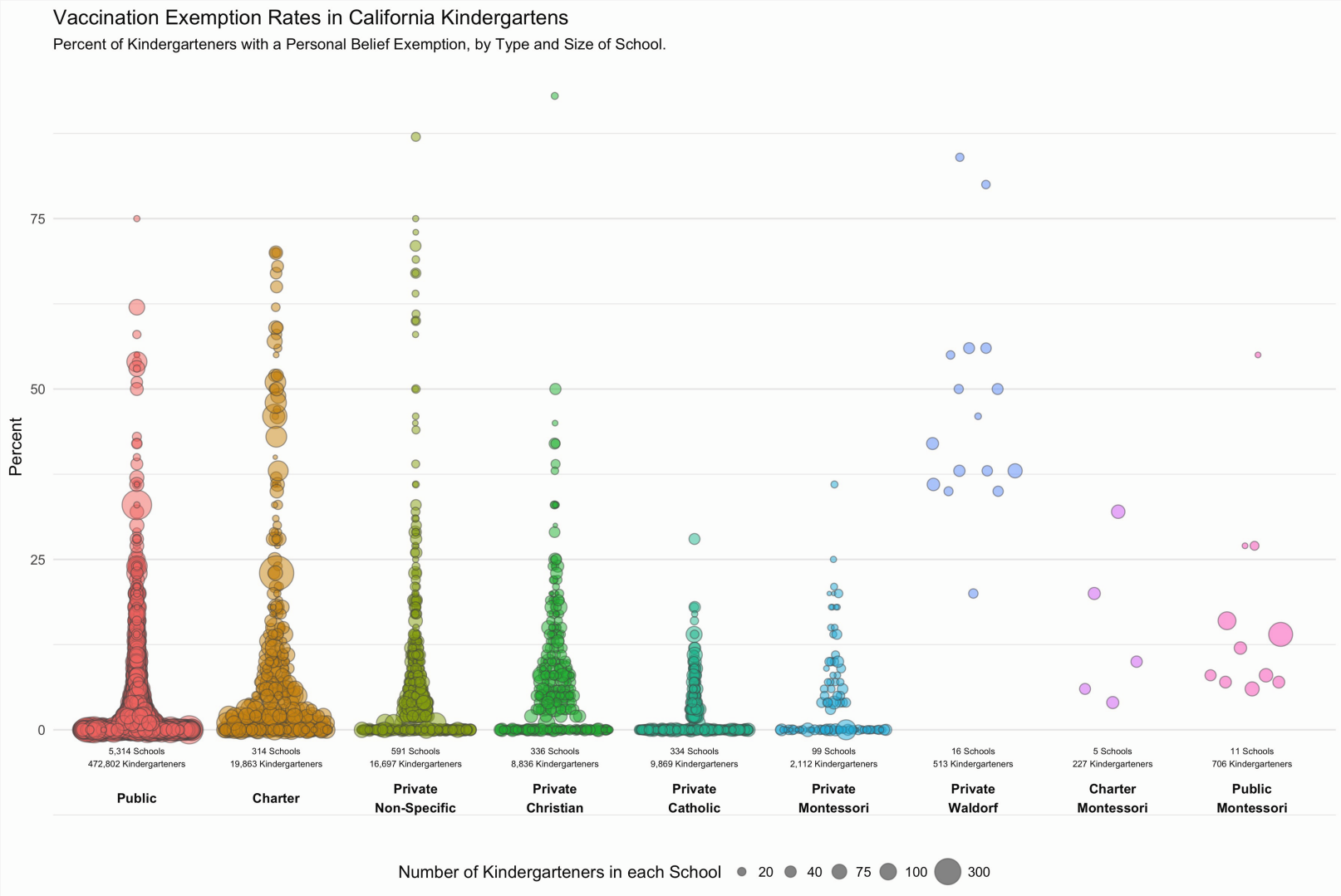
```
studebt ▷
  ggplot(mapping = aes(x = pct/100,
                       y = type_label,
                       fill = Debtrc)) +
  geom_col(color = "gray80") +
  scale_x_continuous(labels =
                       label_percent()) +
  scale_fill_viridis_d() +
  guides(fill =
           guide_legend(reverse = TRUE,
                        title.position = "top",
                        label.position = "bottom",
                        keywidth = 3,
                        nrow = 1)) +
  labs(x = NULL, y = NULL,
       fill = "Amount Owed, in thousands of dollar
       caption = p_caption, title = p_title,
       subtitle = p_subtitle) +
  theme(legend.position = "top",
        plot.title = element_text(size = rel(2.8)
        axis.text = element_text(face = "bold",
                hjust = 1,
                size = rel(2)),
          axis.ticks.length = unit(0, "cm"),
          axis.line = element_blank(),
          panel.grid = element_blank()) ⟶
  p_debt2
```



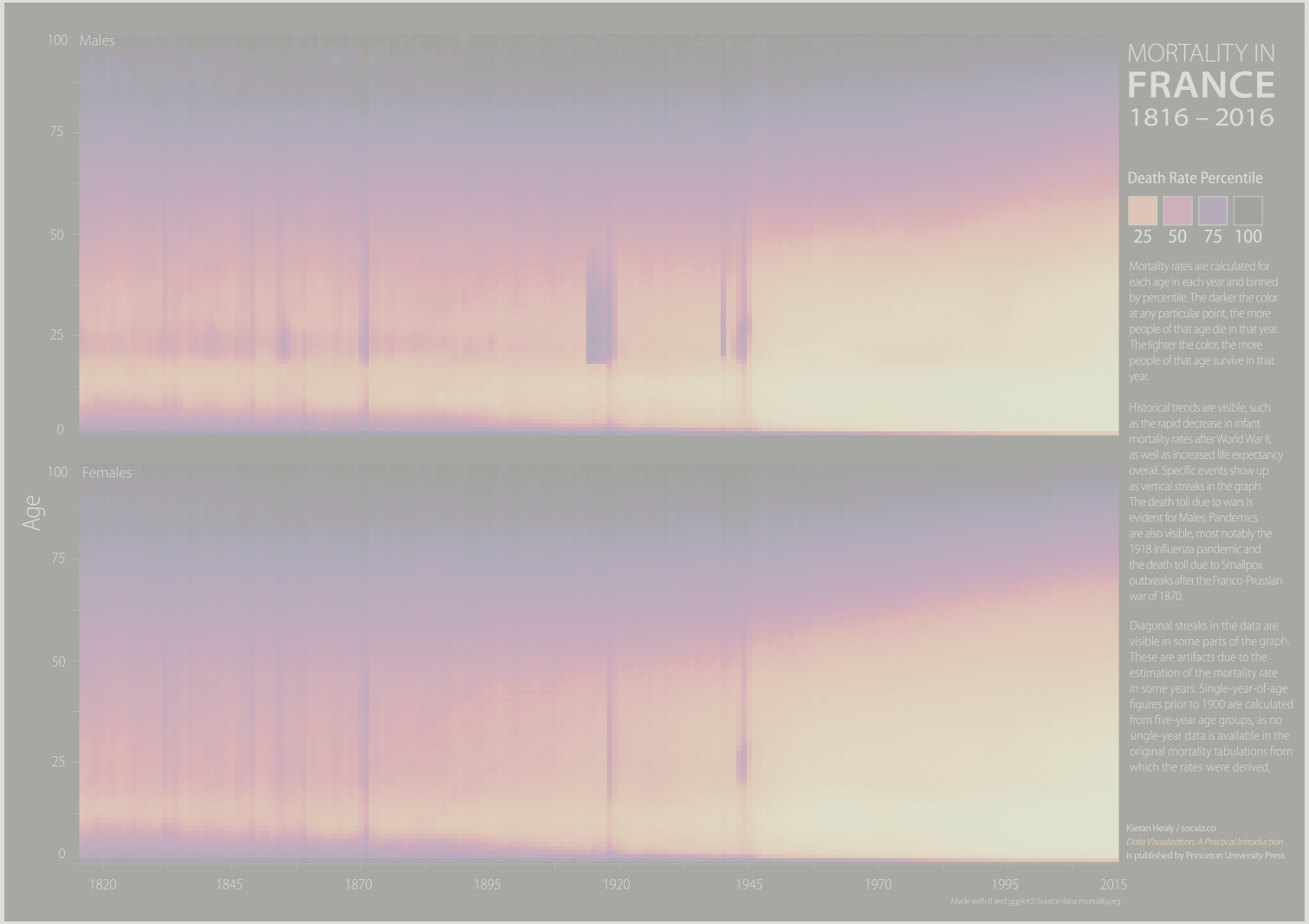
Pies redrawn as sideways-stacked columns

But I want a pony

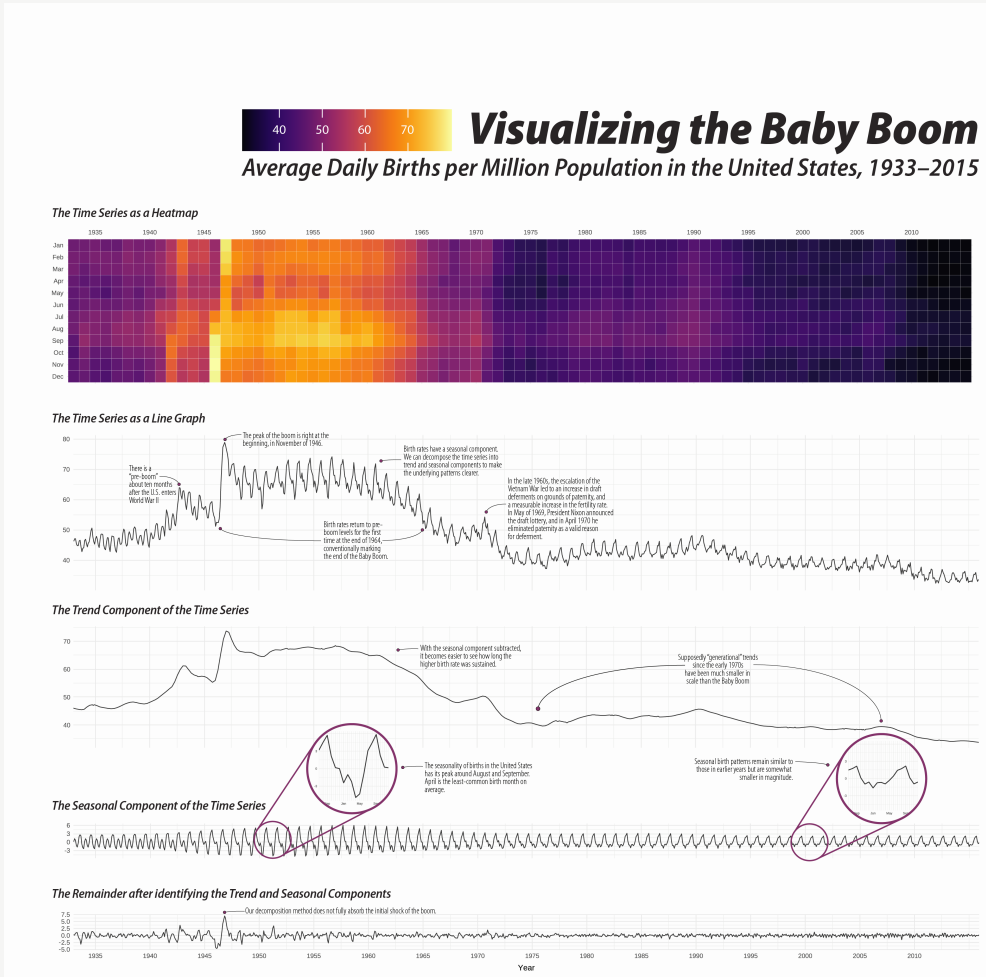
Show ponies


Vaccination Exemption Rates in California Kindergartens
Percent of Kindergarteners with a Personal Belief Exemption, by Type and Size of School.
Percent
75
50
25
0
5,314 Schools
472,802 Kindergarteners
Public
314 Schools
19,863 Kindergarteners
Charter
591 Schools
16,697 Kindergarteners
Private
Non-Specific
336 Schools
8,836 Kindergarteners
Private
Christian
334 Schools
9,869 Kindergarteners
Private
Catholic
99 Schools
2,112 Kindergarteners
Private
Montessori
16 Schools
513 Kindergarteners
Private
Waldorf
5 Schools
227 Kindergarteners
Charter
Montessori
11 Schools
706 Kindergarteners
Public
Montessori
Number of Kindergarteners in each School
20
40
75
100
300

Show ponies


Males
Females
Age
100
75
50
25
0
1820
1845
1870
1895
1920
1945
1970
1995
2015
MORTALITY IN
FRANCE
1816 – 2016
Death Rate Percentile
25 50 75 100
Mortality rates are calculated for each age in each year and binned by percentile. The darker the color at any particular point, the more people of that age die in that year. The lighter the color, the more people of that age survive in that year.
Historical trends are visible, such as the rapid decrease in infant mortality rates after World War II, as well as increased life expectancy overall. Specific events show up as vertical streaks in the graph. The death toll due to wars is evident for Males. Pandemics are also visible, most notably the 1918 Influenza pandemic and the death toll due to Smallpox outbreaks after the Franco-Prussian war of 1870.
Diagonal streaks in the data are visible in some parts of the graph. These are artifacts due to the estimation of the mortality rate in some years. Single-year-of-age figures prior to 1900 are calculated from five-year age groups, as no single-year data is available in the original mortality tabulations from which the rates were derived.
Kieran Healy / socviz.co
Data Visualization: A Practical Introduction
is published by Princeton University Press
Made with R and ggplot2. Source data: mortality.org

Show ponies


40 50 60 70
Visualizing the Baby Boom
Average Daily Births per Million Population in the United States, 1933–2015
The Time Series as a Heatmap
1935 1940 1945 1950 1955 1960 1965 1970 1975 1980 1985 1990 1995 2000 2005 2010
Jan Feb Mar Apr May Jun Jul Aug Sep Oct Nov Dec
The Time Series as a Line Graph
80 70 60 50 40
There is a "pre-boom" about ten months after the U.S. enters World War II
The peak of the boom is right at the beginning, in November of 1946.
Birth rates have a seasonal component. We can decompose the time series into trend and seasonal components to make the underlying patterns clearer.
Birth rates return to pre-boom levels for the first time at the end of 1964, conventionally marking the end of the Baby Boom.
In the late 1960s, the escalation of the Vietnam War led to an increase in draft deferments on grounds of paternity, and a measurable increase in the fertility rate. In May of 1969, President Nixon announced the draft lottery, and in April 1970 he eliminated paternity as a valid reason for deferment.
The Trend Component of the Time Series
70 60 50 40
With the seasonal component subtracted, it becomes easier to see how long the higher birth rate was sustained.
Supposedly "generational" trends since the early 1970s have been much smaller in scale than the Baby Boom
The seasonality of births in the United States has its peak around August and September. April is the least-common birth month on average.
Seasonal birth patterns remain similar to those in earlier years but are somewhat smaller in magnitude.
The Seasonal Component of the Time Series
6 3 0 -3
The Remainder after identifying the Trend and Seasonal Components
7.5 5.0 2.5 0.0 -2.5 -5.0
Our decomposition method does not fully absorb the initial shock of the boom.
1935 1940 1945 1950 1955 1960 1965 1970 1975 1980 1985 1990 1995 2000 2005 2010
Year
Data: United States Census Bureau. Made with R and ggplot by Kieran Healy

OK boomer

The demog package

```
# remotes::install_github("kjhealy/demog")
# library(demog)

okboomer
```

```
## # A tibble: 1,644 × 12
##     year month n_days births total_pop births_pct births_pct_day date
##    <dbl> <dbl>  <dbl>  <dbl>     <dbl>      <dbl>          <dbl> <date>
##  1  1938     1     31  51820  41215000    0.00126           40.6 1938-01-01
##  2  1938     2     28  47421  41215000    0.00115           41.1 1938-02-01
##  3  1938     3     31  54887  41215000    0.00133           43.0 1938-03-01
##  4  1938     4     30  54623  41215000    0.00133           44.2 1938-04-01
##  5  1938     5     31  56853  41215000    0.00138           44.5 1938-05-01
##  6  1938     6     30  53145  41215000    0.00129           43.0 1938-06-01
##  7  1938     7     31  53214  41215000    0.00129           41.6 1938-07-01
##  8  1938     8     31  50444  41215000    0.00122           39.5 1938-08-01
##  9  1938     9     30  50545  41215000    0.00123           40.9 1938-09-01
## 10  1938    10     31  50079  41215000    0.00122           39.2 1938-10-01
## # ℹ 1,634 more rows
## # ℹ 4 more variables: seasonal <dbl>, trend <dbl>, remainder <dbl>,
## #   country <chr>
```

Boomer Line Graph

```
okboomer
```

```
## # A tibble: 1,644 × 12
##     year month n_days births total_pop births_pct births_pct_day date
##    <dbl> <dbl>  <dbl>  <dbl>     <dbl>      <dbl>          <dbl> <date>
##  1  1938     1     31  51820  41215000    0.00126           40.6 1938-01-01
##  2  1938     2     28  47421  41215000    0.00115           41.1 1938-02-01
##  3  1938     3     31  54887  41215000    0.00133           43.0 1938-03-01
##  4  1938     4     30  54623  41215000    0.00133           44.2 1938-04-01
##  5  1938     5     31  56853  41215000    0.00138           44.5 1938-05-01
##  6  1938     6     30  53145  41215000    0.00129           43.0 1938-06-01
##  7  1938     7     31  53214  41215000    0.00129           41.6 1938-07-01
##  8  1938     8     31  50444  41215000    0.00122           39.5 1938-08-01
##  9  1938     9     30  50545  41215000    0.00123           40.9 1938-09-01
## 10  1938    10     31  50079  41215000    0.00122           39.2 1938-10-01
## # ℹ 1,634 more rows
## # ℹ 4 more variables: seasonal <dbl>, trend <dbl>, remainder <dbl>,
## #   country <chr>
```

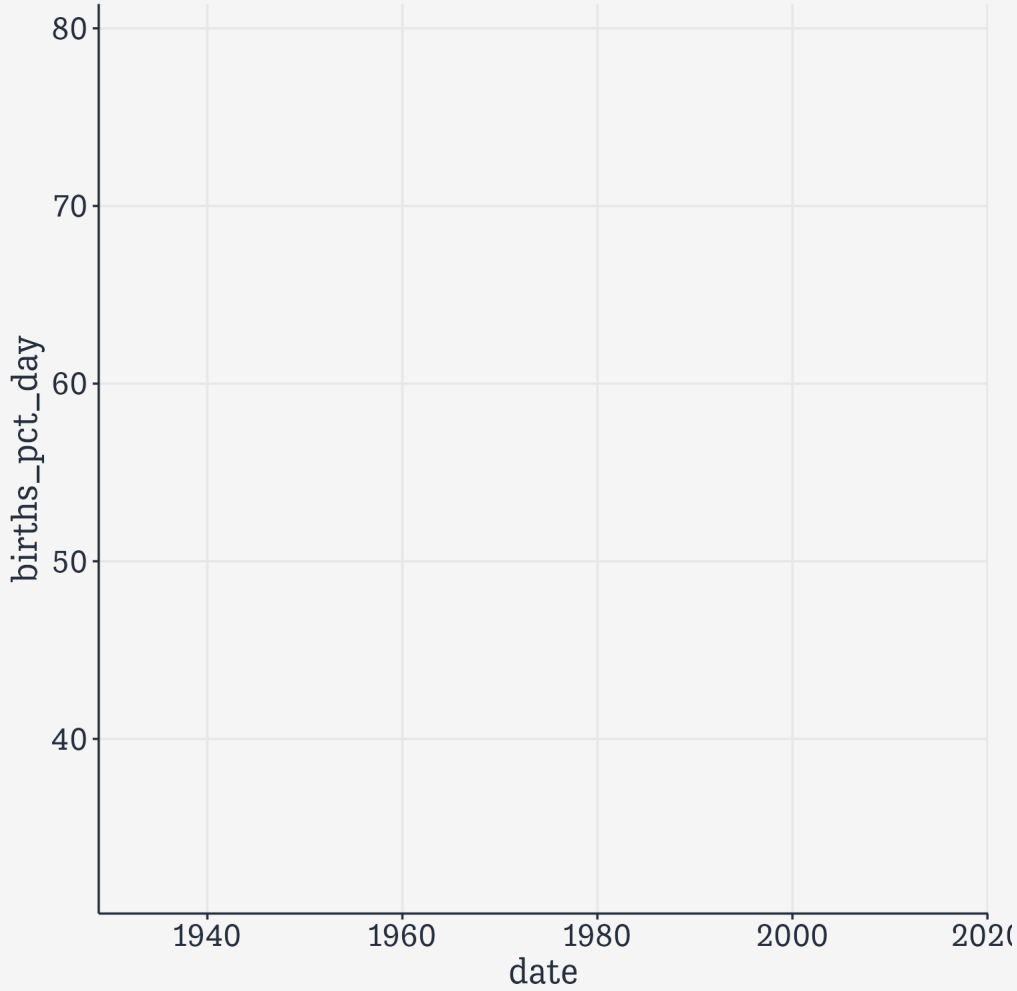
Boomer Line Graph

```
okboomer |>
    filter(country == "United States")
```

```
## # A tibble: 996 × 12
##     year month n_days births total_pop births_pct births_pct_day date
##    <dbl> <dbl>  <dbl>  <dbl>     <dbl>      <dbl>          <dbl> <date>
##  1  1933     1     31 180545 125579000    0.00144           46.4 1933-01-01
##  2  1933     2     28 165986 125579000    0.00132           47.2 1933-02-01
##  3  1933     3     31 183762 125579000    0.00146           47.2 1933-03-01
##  4  1933     4     30 171354 125579000    0.00136           45.5 1933-04-01
##  5  1933     5     31 174811 125579000    0.00139           44.9 1933-05-01
##  6  1933     6     30 169255 125579000    0.00135           44.9 1933-06-01
##  7  1933     7     31 180880 125579000    0.00144           46.5 1933-07-01
##  8  1933     8     31 181856 125579000    0.00145           46.7 1933-08-01
##  9  1933     9     30 167637 125579000    0.00133           44.5 1933-09-01
## 10  1933    10     31 167055 125579000    0.00133           42.9 1933-10-01
## # ℹ 986 more rows
## # ℹ 4 more variables: seasonal <dbl>, trend <dbl>, remainder <dbl>,
## #   country <chr>
```

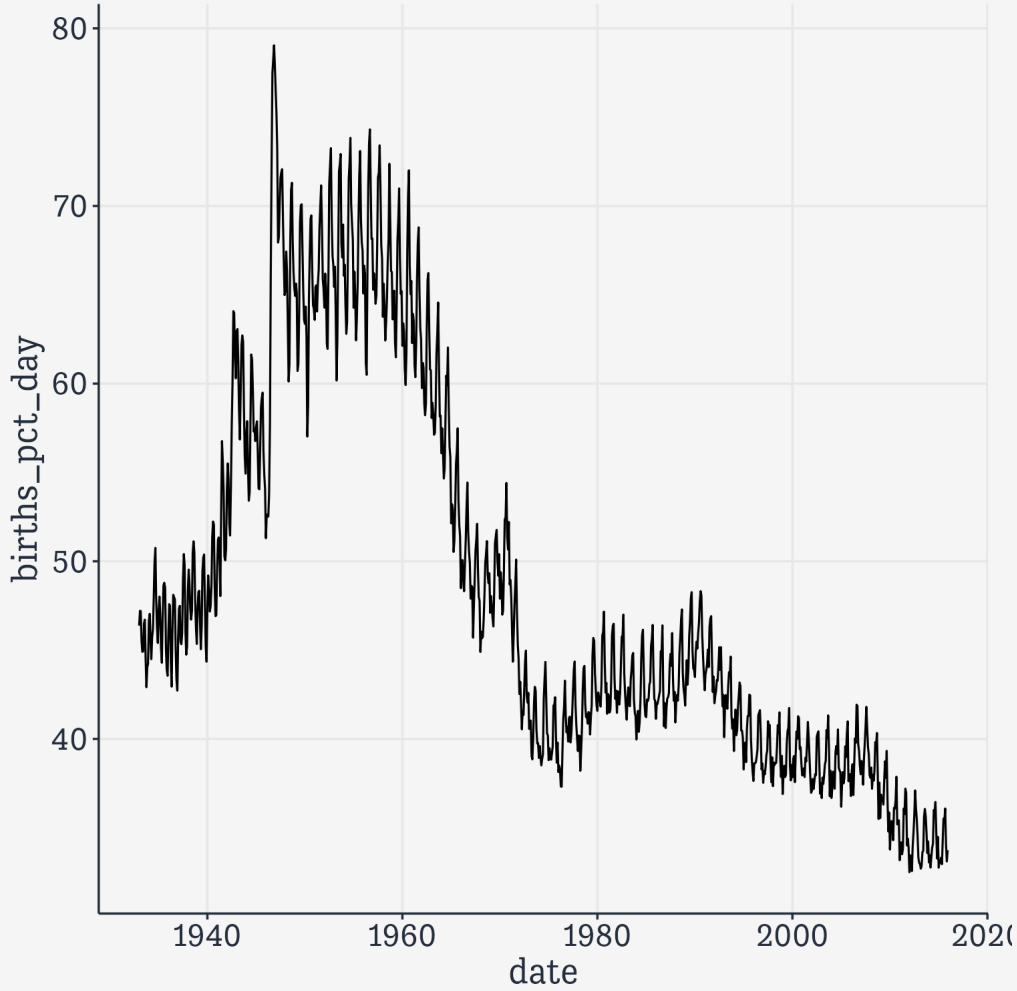
Boomer Line Graph

```
okboomer |>
    filter(country == "United States")  |>
    ggplot(aes(x = date, y = births_pct_day))
```



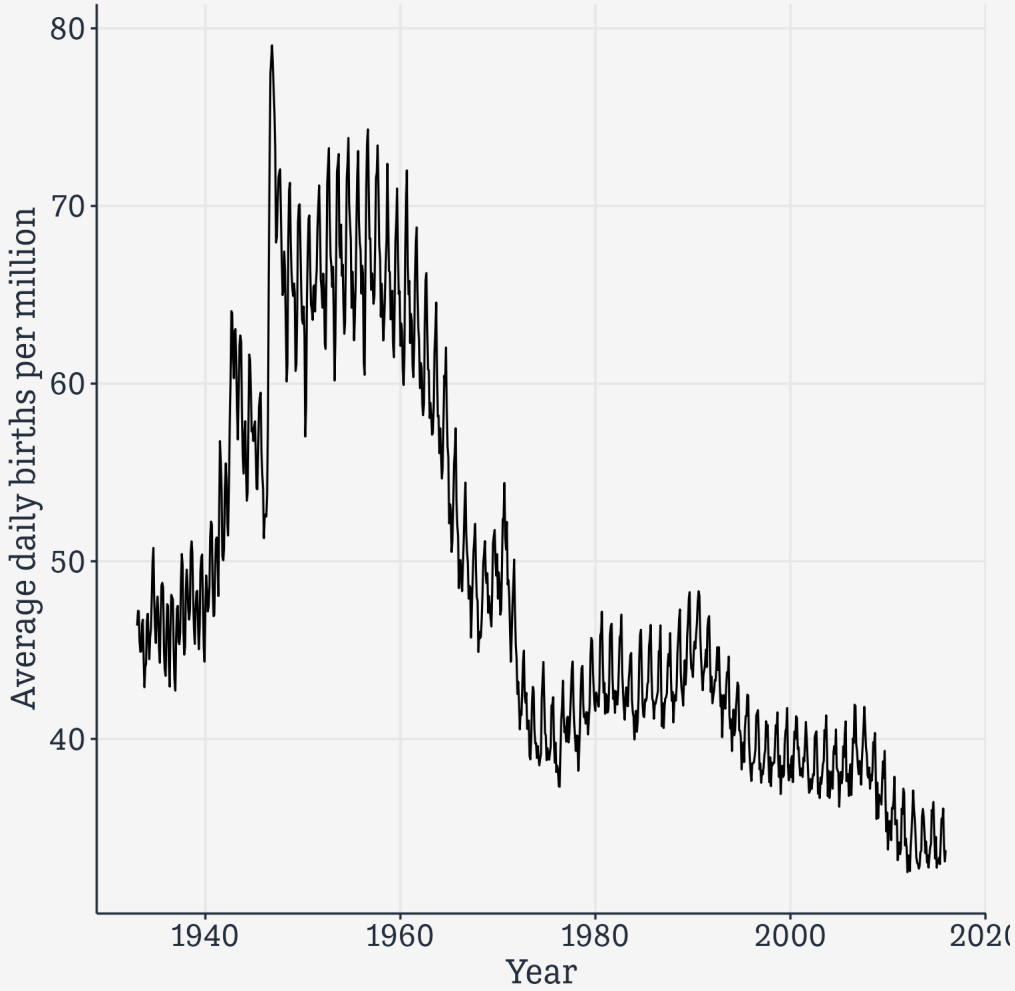
Boomer Line Graph

```
okboomer ▷
    filter(country == "United States")  ▷
    ggplot(aes(x = date, y = births_pct_day)) +
    geom_line(linewidth = 0.5)
```



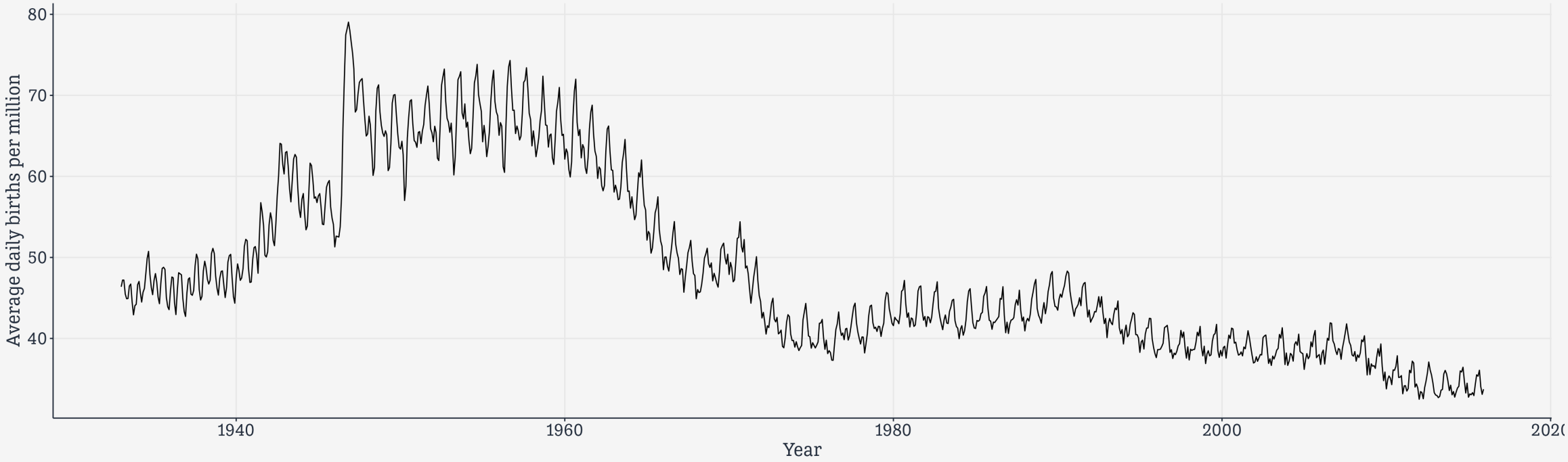
Boomer Line Graph

```
okboomer ▷
    filter(country == "United States")  ▷
    ggplot(aes(x = date, y = births_pct_day)) +
    geom_line(linewidth = 0.5) +
    labs(x = "Year",
         y = "Average daily births per million")
```



Boomer Line Graph

```
okboomer |>
    filter(country == "United States")  |>
    ggplot(aes(x = date, y = births_pct_day)) +
    geom_line(linewidth = 0.5) +
    labs(x = "Year",
         y = "Average daily births per million") +
  p_lineboom
```



The Baby Boom.

Tiled Heatmap

```
okboomer
```

```
## # A tibble: 1,644 × 12
##     year month n_days births total_pop births_pct
##    <dbl> <dbl>  <dbl>  <dbl>     <dbl>      <dbl>
##  1  1938     1     31  51820  41215000    0.00126
##  2  1938     2     28  47421  41215000    0.00115
##  3  1938     3     31  54887  41215000    0.00133
##  4  1938     4     30  54623  41215000    0.00133
##  5  1938     5     31  56853  41215000    0.00138
##  6  1938     6     30  53145  41215000    0.00129
##  7  1938     7     31  53214  41215000    0.00129
##  8  1938     8     31  50444  41215000    0.00122
##  9  1938     9     30  50545  41215000    0.00123
## 10  1938    10     31  50079  41215000    0.00122
## # ℹ 1,634 more rows
## # ℹ 4 more variables: seasonal <dbl>, trend <dbl>,
## #   country <chr>
```

Tiled Heatmap

```
okboomer ▷
    mutate(year_fct =
             factor(year,
                    levels = unique(year),
                    ordered = TRUE),
           month_fct = factor(month,
                              levels = rev(c(1:12)),
                              labels = rev(c("Jan", "Feb", "Mar", "Apr",
                                  "May", "Jun", "Jul", "Aug",
                                  "Sep", "Oct", "Nov", "Dec")),
                              ordered = TRUE))
```

```
## # A tibble: 1,644 × 14
##      year month n_days births total_pop births_pct
##     <dbl> <dbl>  <dbl>  <dbl>     <dbl>      <dbl>
##  1  1938     1     31  51820  41215000    0.00126
##  2  1938     2     28  47421  41215000    0.00115
##  3  1938     3     31  54887  41215000    0.00133
##  4  1938     4     30  54623  41215000    0.00133
##  5  1938     5     31  56853  41215000    0.00138
##  6  1938     6     30  53145  41215000    0.00129
##  7  1938     7     31  53214  41215000    0.00129
##  8  1938     8     31  50444  41215000    0.00122
##  9  1938     9     30  50545  41215000    0.00123
## 10  1938    10     31  50079  41215000    0.00122
## # ℹ 1,634 more rows
## # ℹ 6 more variables: seasonal <dbl>, trend <dbl>,
## #   country <chr>, year_fct <ord>, month_fct <ord>
```

Tiled Heatmap

```
okboomer ▷
    mutate(year_fct =
             factor(year,
                    levels = unique(year),
                    ordered = TRUE),
           month_fct = factor(month,
                              levels = rev(c(1:12)),
                              labels = rev(c("Jan", "Feb", "Mar", "Apr",
                                  "May", "Jun", "Jul", "Aug",
                                  "Sep", "Oct", "Nov", "Dec")),
                              ordered = TRUE)) ▷
    select(year, month, year_fct, month_fct, everything())
```

```
## # A tibble: 1,644 × 14
##     year month year_fct month_fct n_days births to
##    <dbl> <dbl> <ord>    <ord>      <dbl>  <dbl>
##  1  1938     1 1938     Jan           31  51820  4
##  2  1938     2 1938     Feb           28  47421  4
##  3  1938     3 1938     Mar           31  54887  4
##  4  1938     4 1938     Apr           30  54623  4
##  5  1938     5 1938     May           31  56853  4
##  6  1938     6 1938     Jun           30  53145  4
##  7  1938     7 1938     Jul           31  53214  4
##  8  1938     8 1938     Aug           31  50444  4
##  9  1938     9 1938     Sep           30  50545  4
## 10  1938    10 1938     Oct           31  50079  4
## # ℹ 1,634 more rows
## # ℹ 6 more variables: births_pct_day <dbl>, date <
## #   trend <dbl>, remainder <dbl>, country <chr>
```

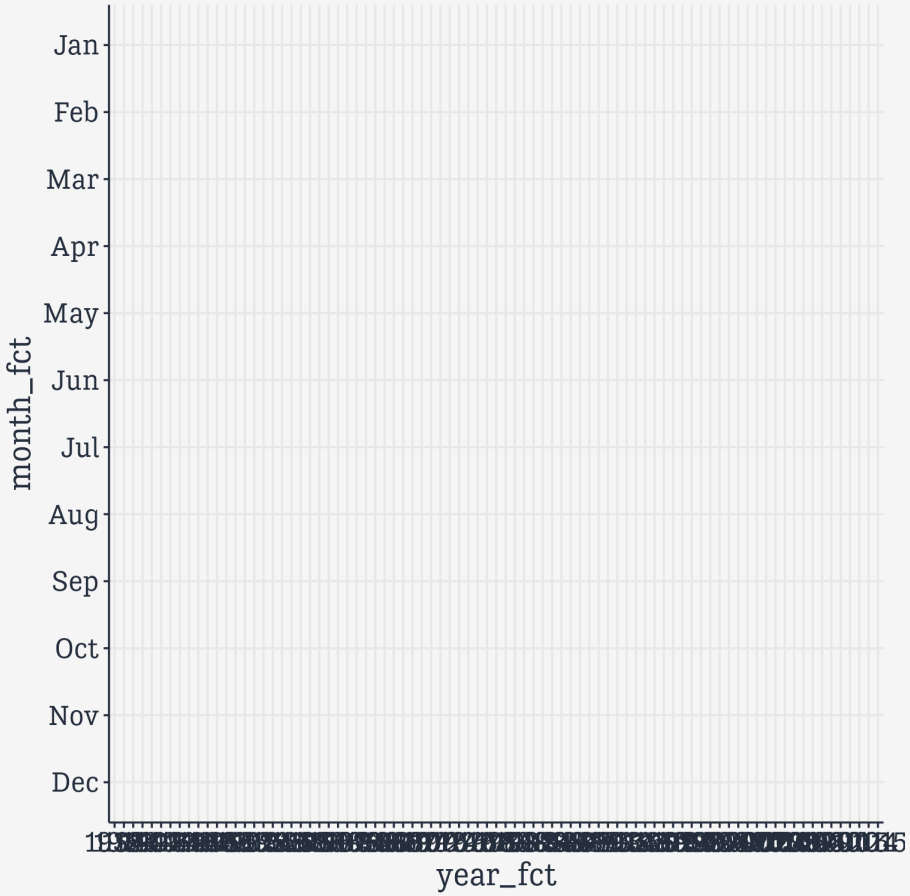
Tiled Heatmap

```
okboomer ▷
    mutate(year_fct =
             factor(year,
                    levels = unique(year),
                    ordered = TRUE),
           month_fct = factor(month,
                              levels = rev(c(1:12)),
                              labels = rev(c("Jan", "Feb", "Mar", "Apr",
                                  "May", "Jun", "Jul", "Aug",
                                  "Sep", "Oct", "Nov", "Dec")),
                              ordered = TRUE)) ▷
    select(year, month, year_fct, month_fct, everything()) ▷
  filter(country == "United States")
```

```
## # A tibble: 996 × 14
##     year month year_fct month_fct n_days births to
##    <dbl> <dbl> <ord>    <ord>      <dbl>  <dbl>
##  1  1933     1 1933     Jan           31 180545 12
##  2  1933     2 1933     Feb           28 165986 12
##  3  1933     3 1933     Mar           31 183762 12
##  4  1933     4 1933     Apr           30 171354 12
##  5  1933     5 1933     May           31 174811 12
##  6  1933     6 1933     Jun           30 169255 12
##  7  1933     7 1933     Jul           31 180880 12
##  8  1933     8 1933     Aug           31 181856 12
##  9  1933     9 1933     Sep           30 167637 12
## 10  1933    10 1933     Oct           31 167055 12
## # ℹ 986 more rows
## # ℹ 6 more variables: births_pct_day <dbl>, date <
## #   trend <dbl>, remainder <dbl>, country <chr>
```

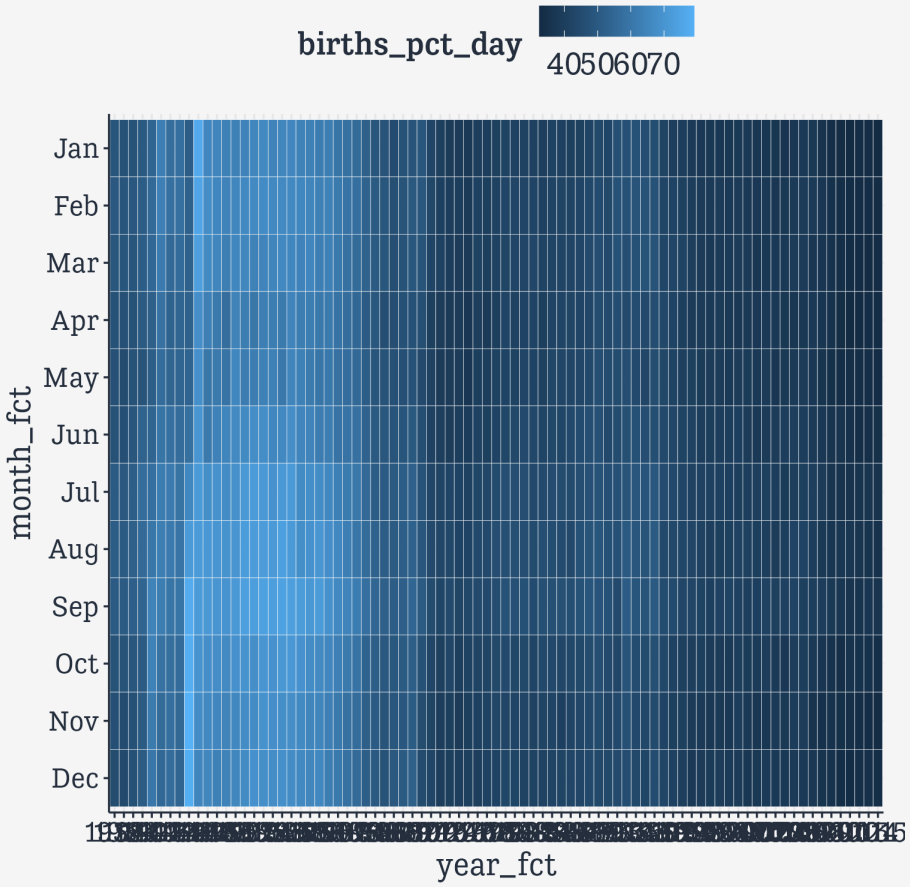
Tiled Heatmap

```
okboomer ▷
    mutate(year_fct =
             factor(year,
                    levels = unique(year),
                    ordered = TRUE),
           month_fct = factor(month,
                              levels = rev(c(1:12)),
                              labels = rev(c("Jan", "Feb", "Mar", "Apr",
                                  "May", "Jun", "Jul", "Aug",
                                  "Sep", "Oct", "Nov", "Dec")),
                              ordered = TRUE)) ▷
    select(year, month, year_fct, month_fct, everything()) ▷
  filter(country == "United States") ▷
    ggplot(aes(x = year_fct, y = month_fct))
```



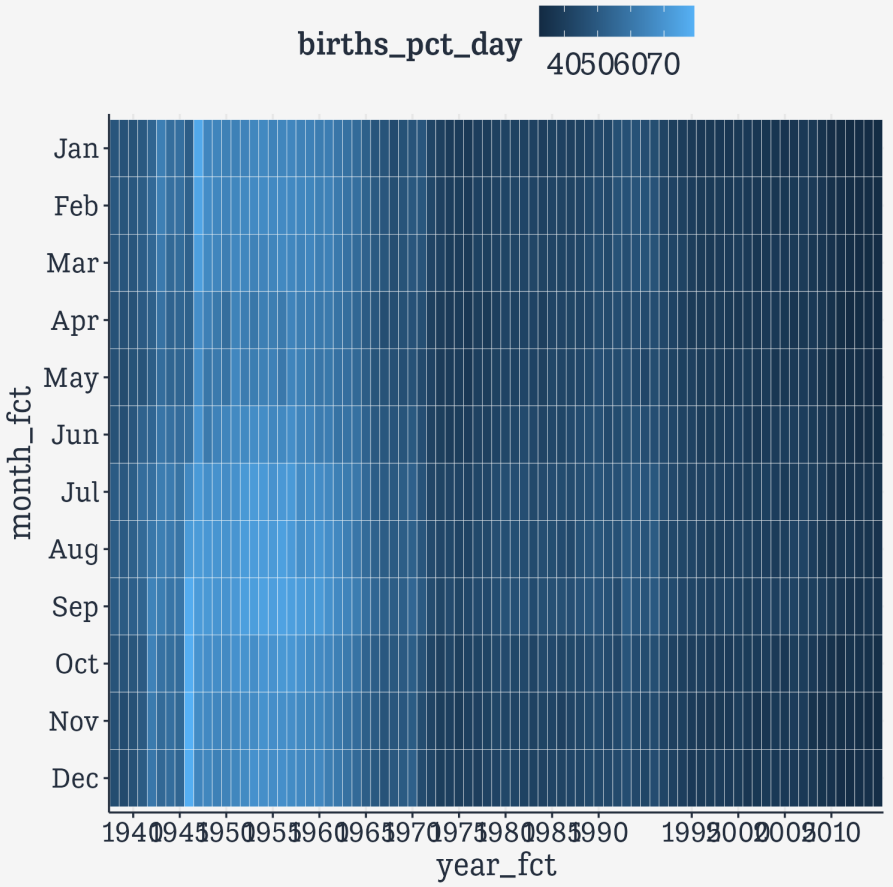
Tiled Heatmap

```
okboomer ▷
    mutate(year_fct =
             factor(year,
                    levels = unique(year),
                    ordered = TRUE),
           month_fct = factor(month,
                              levels = rev(c(1:12)),
                              labels = rev(c("Jan", "Feb", "Mar", "Apr",
                                  "May", "Jun", "Jul", "Aug",
                                  "Sep", "Oct", "Nov", "Dec")),
                              ordered = TRUE)) ▷
    select(year, month, year_fct, month_fct, everything()) ▷
  filter(country == "United States") ▷
    ggplot(aes(x = year_fct, y = month_fct)) +
    geom_tile(mapping = aes(fill = births_pct_day),
              color = "white")
```



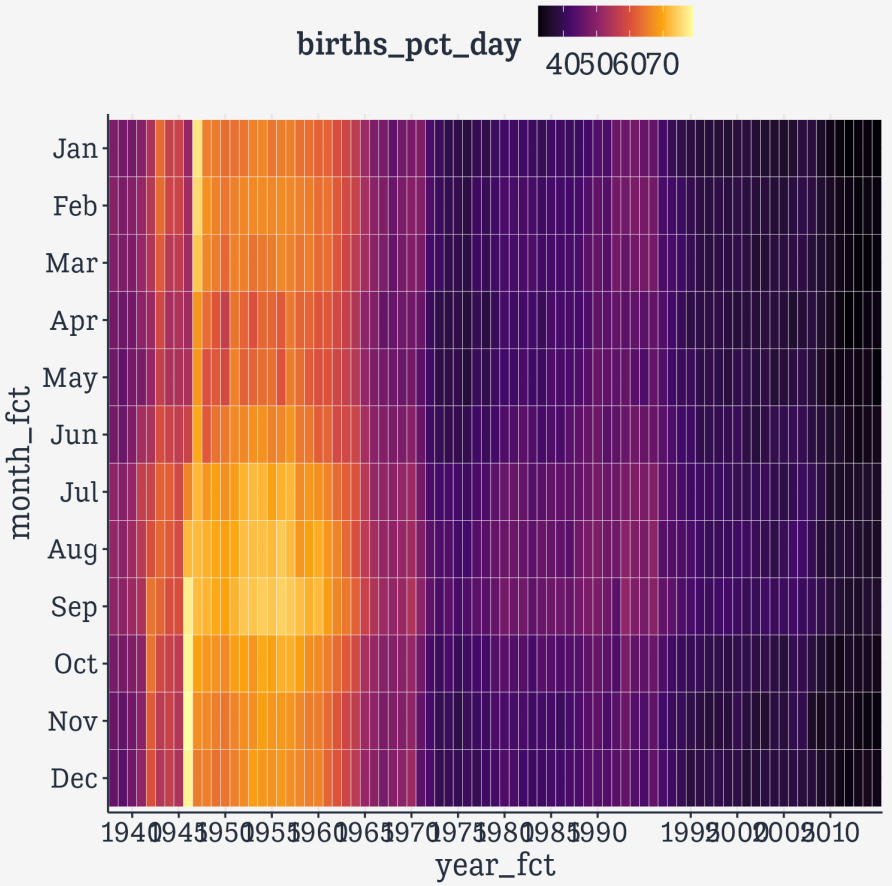
Tiled Heatmap

```
okboomer ▷
    mutate(year_fct =
             factor(year,
                    levels = unique(year),
                    ordered = TRUE),
           month_fct = factor(month,
                              levels = rev(c(1:12)),
                              labels = rev(c("Jan", "Feb", "Mar", "Apr",
                                  "May", "Jun", "Jul", "Aug",
                                  "Sep", "Oct", "Nov", "Dec")),
                              ordered = TRUE)) ▷
    select(year, month, year_fct, month_fct, everything()) ▷
  filter(country == "United States") ▷
    ggplot(aes(x = year_fct, y = month_fct)) +
    geom_tile(mapping = aes(fill = births_pct_day),
              color = "white") +
   scale_x_discrete(breaks = seq(1940, 2010, 5))
```



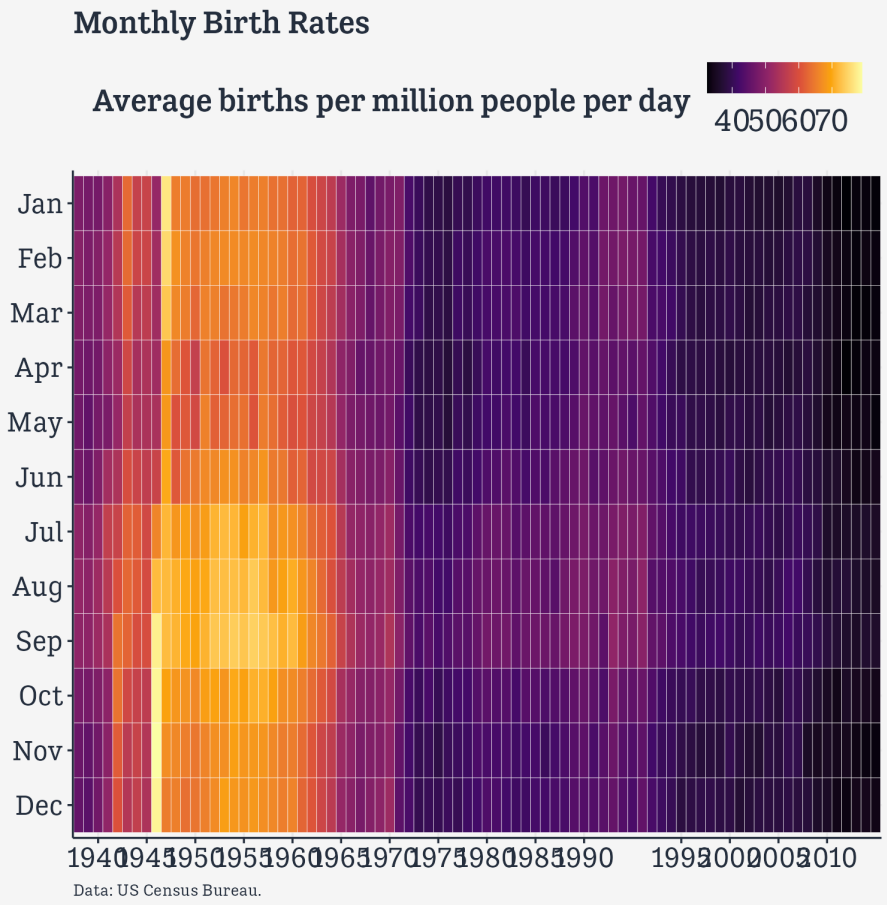
Tiled Heatmap

```
okboomer ▷
    mutate(year_fct =
             factor(year,
                    levels = unique(year),
                    ordered = TRUE),
           month_fct = factor(month,
                              levels = rev(c(1:12)),
                              labels = rev(c("Jan", "Feb", "Mar", "Apr",
                                  "May", "Jun", "Jul", "Aug",
                                  "Sep", "Oct", "Nov", "Dec")),
                              ordered = TRUE)) ▷
    select(year, month, year_fct, month_fct, everything()) ▷
  filter(country == "United States") ▷
    ggplot(aes(x = year_fct, y = month_fct)) +
    geom_tile(mapping = aes(fill = births_pct_day),
              color = "white") +
   scale_x_discrete(breaks = seq(1940, 2010, 5)) +
   scale_fill_viridis_c(option = "B")
```



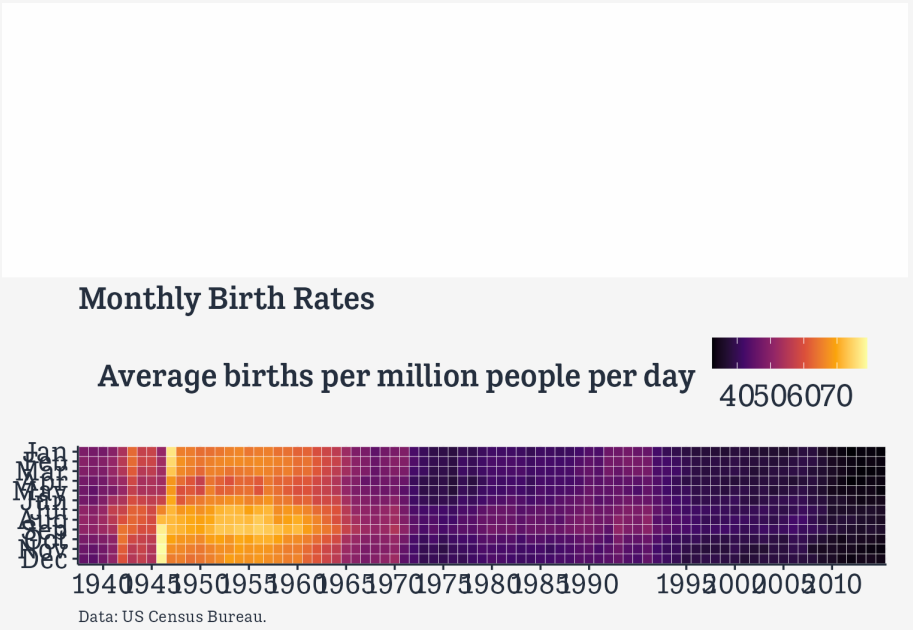
Tiled Heatmap

```
okboomer ▷
    mutate(year_fct =
             factor(year,
                    levels = unique(year),
                    ordered = TRUE),
           month_fct = factor(month,
                              levels = rev(c(1:12)),
                              labels = rev(c("Jan", "Feb", "Mar", "Apr",
                                  "May", "Jun", "Jul", "Aug",
                                  "Sep", "Oct", "Nov", "Dec")),
                              ordered = TRUE)) ▷
    select(year, month, year_fct, month_fct, everything()) ▷
  filter(country == "United States") ▷
    ggplot(aes(x = year_fct, y = month_fct)) +
    geom_tile(mapping = aes(fill = births_pct_day),
              color = "white") +
   scale_x_discrete(breaks = seq(1940, 2010, 5)) +
   scale_fill_viridis_c(option = "B") +
  labs(x = NULL, y = NULL,
       title = "Monthly Birth Rates",
       fill = "Average births per million people per day",
         caption = "Data: US Census Bureau.")
```



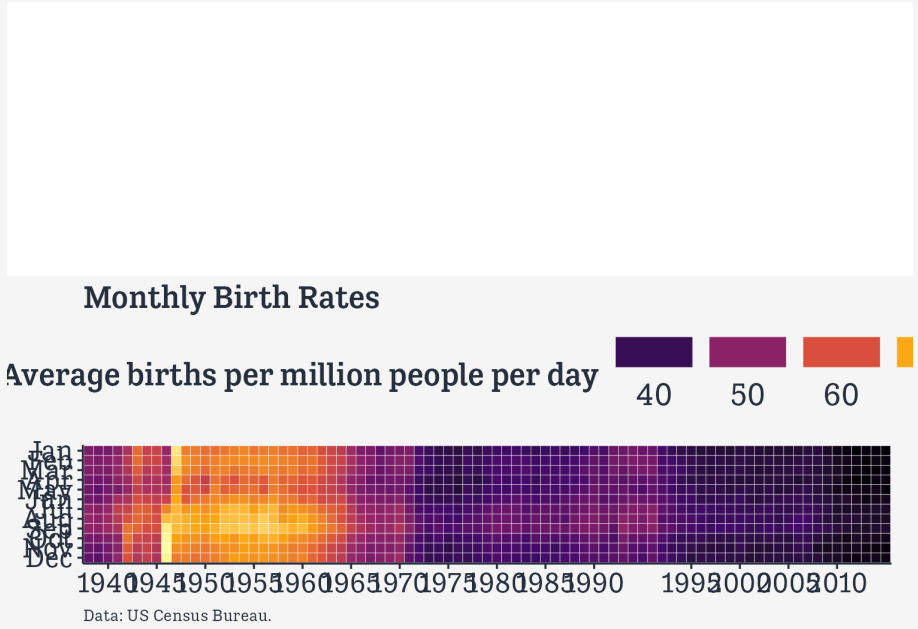
Tiled Heatmap

```
okboomer ▷
    mutate(year_fct =
             factor(year,
                    levels = unique(year),
                    ordered = TRUE),
           month_fct = factor(month,
                              levels = rev(c(1:12)),
                              labels = rev(c("Jan", "Feb", "Mar", "Apr",
                                  "May", "Jun", "Jul", "Aug",
                                  "Sep", "Oct", "Nov", "Dec")),
                              ordered = TRUE)) ▷
    select(year, month, year_fct, month_fct, everything()) ▷
  filter(country == "United States") ▷
    ggplot(aes(x = year_fct, y = month_fct)) +
    geom_tile(mapping = aes(fill = births_pct_day),
              color = "white") +
   scale_x_discrete(breaks = seq(1940, 2010, 5)) +
   scale_fill_viridis_c(option = "B") +
  labs(x = NULL, y = NULL,
       title = "Monthly Birth Rates",
       fill = "Average births per million people per day",
         caption = "Data: US Census Bureau.") +
  coord_fixed()
```



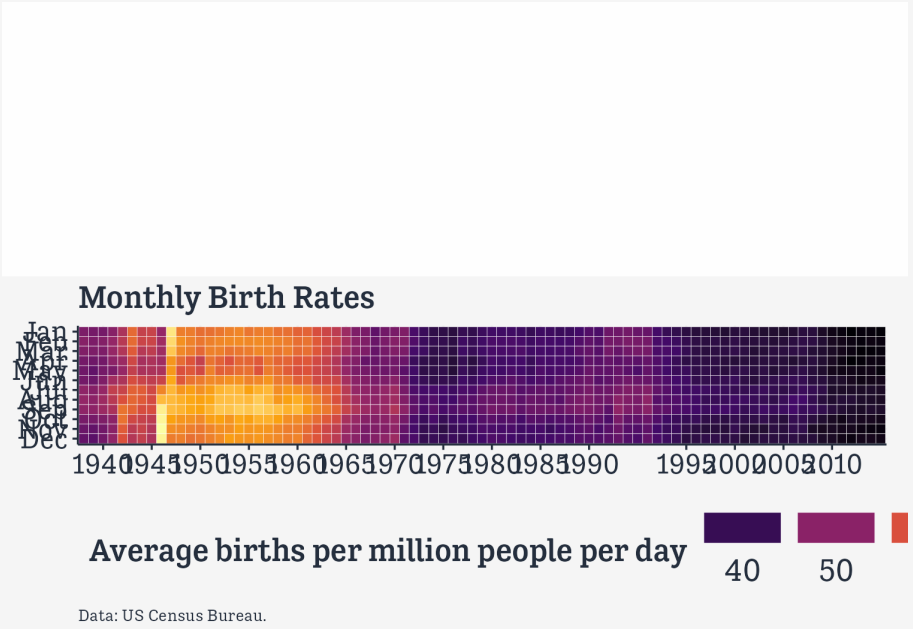
Tiled Heatmap

```
okboomer ▷
    mutate(year_fct =
             factor(year,
                    levels = unique(year),
                    ordered = TRUE),
           month_fct = factor(month,
                              levels = rev(c(1:12)),
                              labels = rev(c("Jan", "Feb", "Mar", "Apr",
                                  "May", "Jun", "Jul", "Aug",
                                  "Sep", "Oct", "Nov", "Dec")),
                              ordered = TRUE)) ▷
    select(year, month, year_fct, month_fct, everything()) ▷
  filter(country == "United States") ▷
    ggplot(aes(x = year_fct, y = month_fct)) +
    geom_tile(mapping = aes(fill = births_pct_day),
              color = "white") +
   scale_x_discrete(breaks = seq(1940, 2010, 5)) +
   scale_fill_viridis_c(option = "B") +
  labs(x = NULL, y = NULL,
       title = "Monthly Birth Rates",
       fill = "Average births per million people per day",
         caption = "Data: US Census Bureau.") +
  coord_fixed() +
  guides(fill = guide_legend(keywidth = 3,
                     label.position = "bottom"))
```



Tiled Heatmap

```
okboomer ▷
    mutate(year_fct =
             factor(year,
                    levels = unique(year),
                    ordered = TRUE),
           month_fct = factor(month,
                              levels = rev(c(1:12)),
                              labels = rev(c("Jan", "Feb", "Mar", "Apr",
                                  "May", "Jun", "Jul", "Aug",
                                  "Sep", "Oct", "Nov", "Dec")),
                              ordered = TRUE)) ▷
    select(year, month, year_fct, month_fct, everything()) ▷
  filter(country == "United States") ▷
    ggplot(aes(x = year_fct, y = month_fct)) +
    geom_tile(mapping = aes(fill = births_pct_day),
              color = "white") +
   scale_x_discrete(breaks = seq(1940, 2010, 5)) +
   scale_fill_viridis_c(option = "B") +
  labs(x = NULL, y = NULL,
       title = "Monthly Birth Rates",
       fill = "Average births per million people per day",
         caption = "Data: US Census Bureau.") +
  coord_fixed() +
  guides(fill = guide_legend(keywidth = 3,
                    label.position = "bottom")) +
  theme(legend.position = "bottom",
        legend.justification = "left")
```

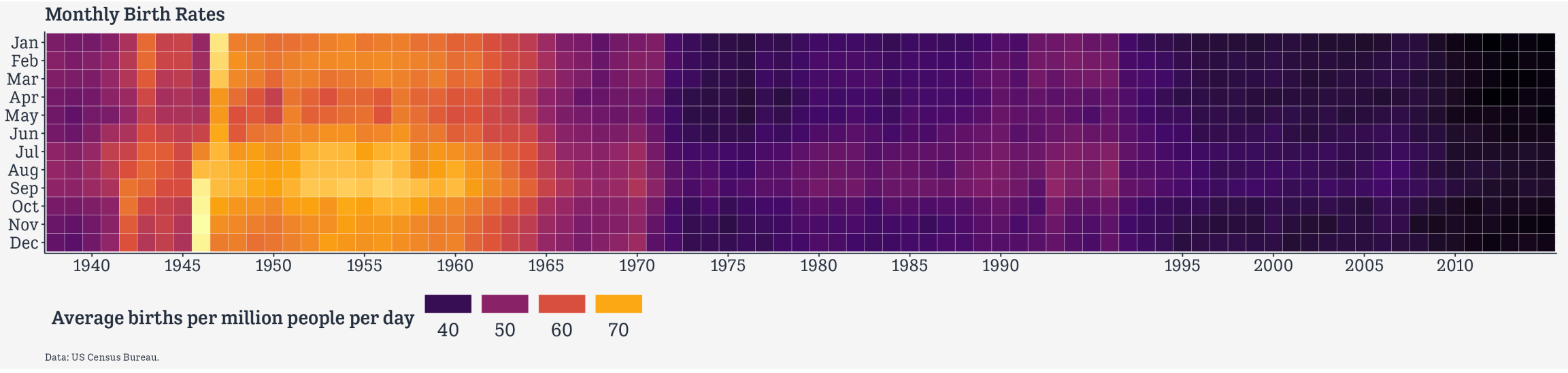


Tiled Heatmap

```
okboomer ▷
    mutate(year_fct =
             factor(year,
                    levels = unique(year),
                    ordered = TRUE),
           month_fct = factor(month,
                              levels = rev(c(1:12)),
                              labels = rev(c("Jan", "Feb", "Mar", "Apr",
                                  "May", "Jun", "Jul", "Aug",
                                  "Sep", "Oct", "Nov", "Dec")),
                              ordered = TRUE)) ▷
    select(year, month, year_fct, month_fct, everything()) ▷
  filter(country == "United States") ▷
    ggplot(aes(x = year_fct, y = month_fct)) +
    geom_tile(mapping = aes(fill = births_pct_day),
              color = "white") +
   scale_x_discrete(breaks = seq(1940, 2010, 5)) +
   scale_fill_viridis_c(option = "B") +
  labs(x = NULL, y = NULL,
       title = "Monthly Birth Rates",
       fill = "Average births per million people per day",
         caption = "Data: US Census Bureau.") +
  coord_fixed() +
  guides(fill = guide_legend(keywidth = 3,
                    label.position = "bottom")) +
  theme(legend.position = "bottom",
        legend.justification = "left") →
  p_tileboom
```

Tiled Heatmap

```
okboomer ▷
    mutate(year_fct =
             factor(year,
                    levels = unique(year),
                    ordered = TRUE),
           month_fct = factor(month,
                              levels = rev(c(1:12)),
                              labels = rev(c("Jan", "Feb", "Mar", "Apr",
                                  "May", "Jun", "Jul", "Aug",
                                  "Sep", "Oct", "Nov", "Dec")),
                              ordered = TRUE)) ▷
    select(year, month, year_fct, month_fct, everything()) ▷
  filter(country == "United States") ▷
    ggplot(aes(x = year_fct, y = month_fct)) +
    geom_tile(mapping = aes(fill = births_pct_day),
              color = "white") +
   scale_x_discrete(breaks = seq(1940, 2010, 5)) +
   scale_fill_viridis_c(option = "B") +
  labs(x = NULL, y = NULL,
       title = "Monthly Birth Rates",
       fill = "Average births per million people per day",
         caption = "Data: US Census Bureau.") +
  coord_fixed() +
  guides(fill = guide_legend(keywidth = 3,
                    label.position = "bottom")) +
  theme(legend.position = "bottom",
        legend.justification = "left") →
  p_tileboom
```



The Baby Boom as a tiled temporal heatmap

Iteration:
Many figures at once

We'll use a separate document for this one.