The MockUp

Three years of TidyTuesday

The future holds a deeper focus on inclusivity.

META DATA VISUALIZATION TIDYTUESDAY GGPLOT2

AUTHOR
Thomas Mock

PUBLISHED
April 1, 2021

The `#TidyTuesday` project has been active since April 1st of 2018, so today is the 3 year anniversary!

> As this project was borne out of the R4DS Online Learning Community and the R for Data Science textbook, an emphasis was placed on understanding how to summarize and arrange data to make meaningful charts with `ggplot2`, `tidyr`, `dplyr`, and other tools in the `tidyverse` ecosystem. However, any code-based methodology is welcome - just please remember to share the code used to generate the results.

There was even a paper published on TidyTuesday!

Over the past 3 years, 1000s of people have contributed their submissions and "shown their work" with open source R code! Since the primary way people engage with TidyTuesday is via Twitter submissions, I have a weekly tweet collector that stores and uploads the Tweets to the TidyTuesday GitHub.

This allows us to do some basic counting of a few different parameters over time. However, before we get to the details I also want to get into a mistake we've made and have partially rectified. Thanks to efforts by Dr. Liz Hare and Dr. Silvia Canelon, I was made aware that less than 3% of submissions

The MockUp

I recommend listening to their talk at CSV conf 2021 on May 5th at 10:20am EST for the full details, and some advice on how to improve our accessibility.

If you're not aware, alt text is the primary way that screen reader software used by low vision or blind users interact with many web pages or other software. Per the American Foundation for the Blind:

> Screen readers are software programs that allow blind or visually impaired users to read the text that is displayed on the computer screen with a speech synthesizer or braille display.

color blindness, it doesn't capture the whole story. Low vision, altered vision, or motor impairment as opposed to clinically-defined blindness is even more prevalent. I think that the thread by Frank Elavsky covers more details that are worth reading, and emphasizing that inclusivity doesn't stop at color blind safe palettes.

> **Frank** @FrankElavsky · Jan 18, 2021
> Replying to @FrankElavsky
> So now what??
>
> My solution is simple:
> Prioritize more discourse, research, guides, palettes, and tools that help with other functional disabilities and disabled communities.
>
> And please stop writing about colorblindness in visualization. We have enough of those blogs already!
>
> **Frank**
> @FrankElavsky · **Follow**
>
> Resources we could use more of (1/?):
> Low vision (~30% of all people):
> - High contrast text
> - High contrast elements
> - Using texture, shape, units
> - Designing with zoom/magnification
> - Using Hierarchy and Focus
> - Using annotations or guides
>
> 8:33 PM · Jan 18, 2021
>
> 854 Reply Copy link
>
> **Read 7 replies**

Ultimately, there is a need to move beyond simply thinking about color-blindness, and making sure to include things like alt text, emphasizing contrast, and generally make our data visualization more engaging for all

Alt text for graphs

The Data Visualization Society has a great article by Amy Cesal on Writing Alt Text for Data Visualization.

> You probably can't write text that conveys the entire meaning of a chart. But, that doesn't mean that you shouldn't try.

Ultimately, this is a hard problem, but just as Amy says - that doesn't mean we can't or shouldn't try to do our best to describe our graphs for folks who aren't able to see all the details.

She recommends alt text that includes:

- Chart type
- The type of data
- Reason for including the chart
- Link to the data source

The goal here is to add a rich description of the PURPOSE of the graph, which is ultimately the goal of data visualization - telling a story with data.

I have a brief script using glue that allows you to write this style of alt text.

```
write_alt_text <-       function(
  chart_type,
  type_of_data,
  reason   ,
  source
  )       {
  glue    ::      glue    (
    "{chart_type} of {type_of_data} where {reason}. \n\nData s
    )
}
```

The MockUp

```
  "Bar Chart",
  "tweets from the past week tagged with the TidyTuesday hash
  "Tuesday is the most popular day to post tweets, although al
  "the {rtweet} package."
  )
```

```
Bar Chart of tweets from the past week tagged with the
TidyTuesday hashtag where Tuesday is the most popular day to
post tweets, although about 20-30 tweets are posted every
day.

Data source from the {rtweet} package.
```

This can be easily added to Twitter-based images via the various clients, and will also be able to be uploaded via rtweet in the future. Twitter has a full guide on how to add image alt text.

> When you Tweet photos using the Twitter app for iOS or Android, or on twitter.com, you have the option to compose a description of the images so the content is accessible to more people, including those who are blind or low-vision.
>
> Good image descriptions are concise and descriptive, helping people understand what's happening in an image.

Without alt text, most images on Twitter will just default to reporting "Image" as the alt text for any images you display. You could imagine that while a picture is worth a 1000 words, that the word "image" is far from helpful in describing said image.

Furthermore, Penn State has a chart description guide that covers charts and accessibility.

> Generally speaking an ALT tag cannot do justice to a complex chart.

The MockUp

> This serves multiple audiences because a chart can show trends, but a table can provide exact data for those who are interested.

I love this idea of providing the raw data in a table or a source in addition to the image itself. However, I think that also making sure to write a good text description is better than just sharing the data or table alone.

Lastly, to help you remember to always include alt text for images and gifs on Twitter, you can add extensions like Chrome's Twitter Required Alt Text. This actually prevents you from uploading images WITHOUT alt text.

Future effort

While I've made some changes to my personal workflow and including alt text in my "automated" tweets around `#TidyTuesday`, I hope that we as the R community and more specifically the `#TidyTuesday` community can make it a priority to add more alt text to our graphs and be more inclusive of the wider community.

While I am excited about the work we have done in the past, we obviously have a lot to do in the future to make this a reality. As part of this, I'm going to be walking through some of the past submission data and writing out example alt text, as well as including tables.

For RMarkdown docs/blogs, you can add alt text very easily!

First off, the basic syntax provides an area for alt text, detailed in the RMarkdown Cookbook.

```
![an informative text](path/to/image.png)
```

OR with `knitr::include_graphics()`, which is what I've done for the

The MockUp

```
knitr::include_graphics('figures/my_image.png')
```

#TidyTuesday data

With this in mind, we can walk through the past TidyTuesday posts, create some graphics that show the changes, write some alt text that describes said graphs, and include tables to further "tell our story". All of the historical data/tweets are collected in the TidyTuesday repo.

```
library (       tidyverse)
library (       lubridate)

raw_url   <-        "https://raw.githubusercontent.com/rfordata

raw_df    <-        read_csv(        raw_url  )
```

There are MANY tweets!

```
raw_df    %>%
  distinct(       status_id)        %>%
  count    (       )
```

```
# A tibble: 1 × 1
      n
  <int>
1 17284
```

And there are still many tweets if we filter to only include Tweets with a few key words of interest to R/Data visualization.

```
raw_df    %>%
  filter   (       str_detect(       tolower (       text
```

The MockUp

```
# A tibble: 1 × 1
      n
  <int>
1 13076
```

Lastly, there are thousands of unique contributors over the past 3 years.

```
raw_df   %>%
  filter  (       str_detect(       tolower (       text
  distinct(       screen_name)       %>%
  count   (       )
```

```
# A tibble: 1 × 1
      n
  <int>
1  2667
```

Data visualizations

We can create some graphics based on this data. I'm going to create some summary datasets, and then get the top weeks for labeling.

```
sum_df   <-       raw_df   %>%
  filter  (       str_detect(       tolower (       text
  mutate  (
    created_date =       lubridate::      as_date (       c
    year =       year    (       created_date)       ,
    week =       week    (       created_date)
  )       %>%
  filter  (       year     %in%     2018    :       2021
  count   (       year    , week    )       %>%
```

The MockUp

```
    week =         if_else (        year      >         2018
    year =         factor  (        year     , levels =
    colour =         case_when(
      year      ==        2021      ~         "#FF2B4F",
      year      ==        2020      ~         "#3686d3",
      year      ==        2019      ~         "#003399",
      year      ==        2018      ~         "#88398a",
      TRUE      ~         "gray80"
    )
  )         %>%
  ungroup (         )

top_weeks <-         sum_df    %>%
  group_by(         year     )         %>%
  arrange (         desc     (        roll_n   )         )
  mutate  (         mean =          mean     (         n         )
  slice   (         1        )         %>%
  ungroup (         )
```

Growth chart by year

The plotting code is below, and the included alt text:

> A line chart of TidyTuesday-tagged tweets across the past 3 years, where there is large year over year growth. 2021 is on pace to exceed the total tweet counts from 2020.

```
tt_plot  <-        ggplot  (
  sum_df  ,
  aes     (
    x =          week     ,
    y =          roll_n   ,
    color =          colour   ,
    group =          year
```

The MockUp

```
geom_point(
  data =         top_weeks,
  aes     (        col =         colour   )        ,
  size =         2.5      ,
  stroke =         1
)         +
geom_text(
  data =         top_weeks,
  aes     (        label =         year     )        ,
  size =         8        ,
  hjust =          c        (        1        , 1        , 1
  nudge_y =          50       ,
  vjust =         0
)         +
geom_hline(         yintercept =         0         , size =
scale_y_continuous(
  breaks =         seq      (        0        , 5500     , by =
  limits =         c        (        0        , 5500     )
)         +
scale_x_continuous(
  breaks =         c        (        seq      (        0
  limits =         c        (        0        , 53       )
)         +
scale_color_identity(         aesthetics =         c        (
labs     (
  x =          "\nWeek Number",
  y =          "Cumulative Tweets\n",
  caption =          "Data: rtweet | Plot: @thomas_mock",
  title =          "Cumulative tweets for #TidyTuesday by yea
  subtitle =          "Note that Week 1 of 2018 started in Ap
)         +
tomtom   ::       theme_538(         )         +
theme    (
  legend.position =         c        (        0.1      , 0.8
  legend.background =          element_blank(         )

  legend.title =         element_blank(         )        ,
```

The MockUp

```
  )

ggsave  (
  "tt_tweets.png",
  tt_plot ,
  device =        ragg    ::      agg_png (
    width =        10       ,
    height =        10       ,
    units =        "in"     ,
    scaling =         0.8      ,
    res =        500
  )
)
```

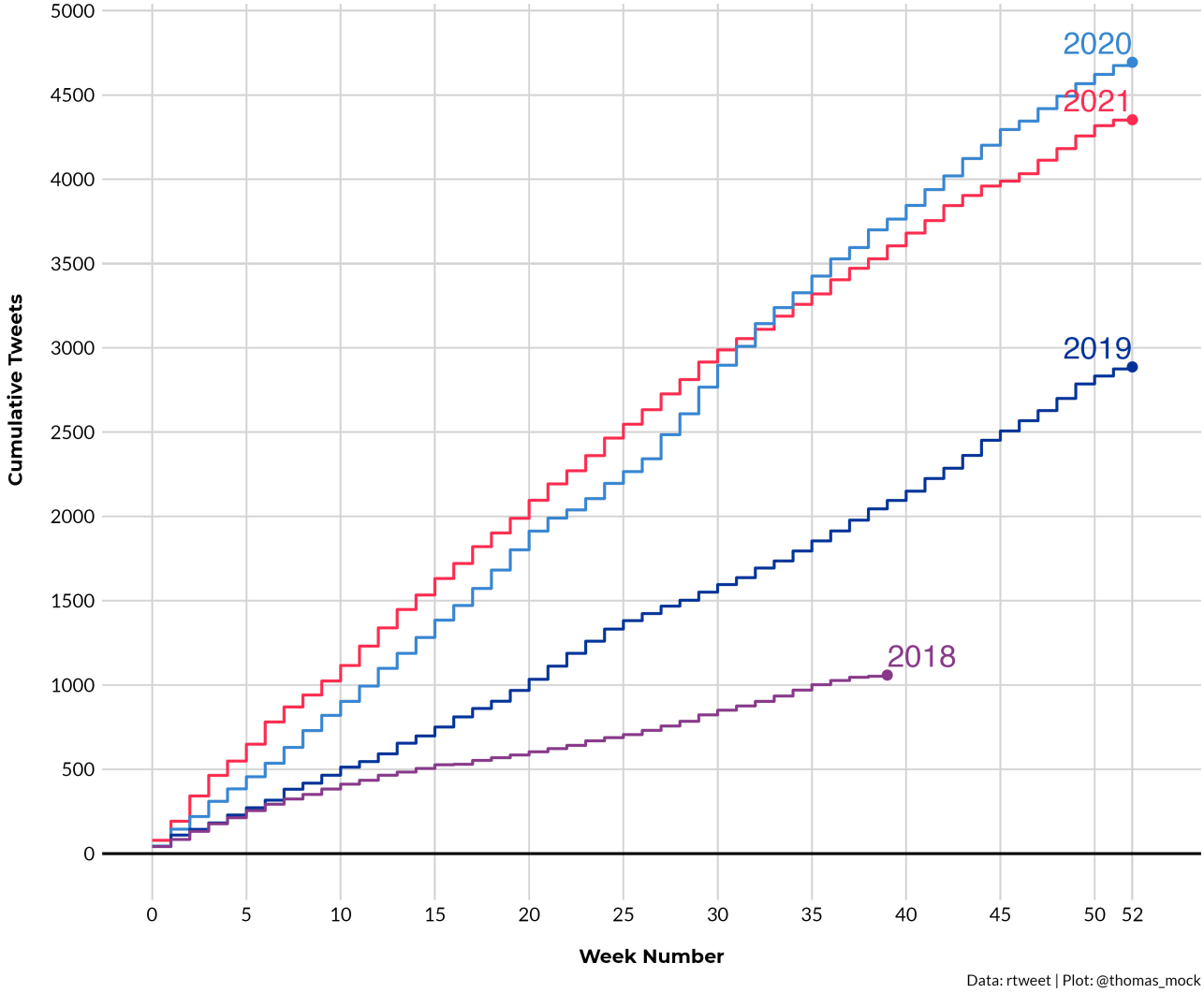
```
Saving 12.5 x 12.5 in image
```

```
knitr   ::      include_graphics(       "tt_tweets.png"
)
```

Cumulative tweets for #TidyTuesday by year

Note that Week 1 of 2018 started in April & tweets must contain: 'rstats, code, plot, graph, viz, data or tidyverse'

The MockUp



A line chart of TidyTuesday-tagged tweets across the past 3 years, where there is large year over year growth. 2021 is on pace to exceed the total tweet counts from 2020.

Data sourced from the TidyTuesday repo.

```
library ( gt )

top_weeks %>%
  select ( year : roll_n , mean )
  gt ( ) %>%
  fmt_number( columns = vars ( roll_n
  cols_label(
    week = html ( "Week<br>Number") ,

    n = html ( "Tweets in<br>last week")
    roll_n = html ( "Rolling<br>Tweet Count"
```

The MockUp

```
    title =          md       (          "**Summary of TidyTuesday
  )         %>%
  espnscrapeR::       gt_theme_538(         )         %>%
  tab_source_note(
    html     (         "<strong>Data Source: </strong><a href='h
  )
```

```
Warning: `columns = vars(...)` has been deprecated in gt
0.3.0:
* please use `columns = c(...)` instead

Warning: `columns = vars(...)` has been deprecated in gt
0.3.0:
* please use `columns = c(...)` instead
```

Summary of TidyTuesday Tweets by year

YEAR	WEEK NUMBER	TWEETS IN LAST WEEK	ROLLING TWEET COUNT	MEAN
2021	52	2	4,353	82
2020	52	19	4,694	89
2019	52	12	2,887	54
2018	39	7	1,059	26

Data Source: TidyTuesday Repo

Peak weeks by year

The next graphic is intended to show the peak number of tweets recorded for each week, separated out by year. The alt text is included below the graphc, but also described as so:

> A bar chart of TidyTuesday-tagged tweets across the past 3 years,
>
> split into small multiples by year, where each year shows higher weekly peaks over the same week as last year. 2021 is on pace to

The MockUp

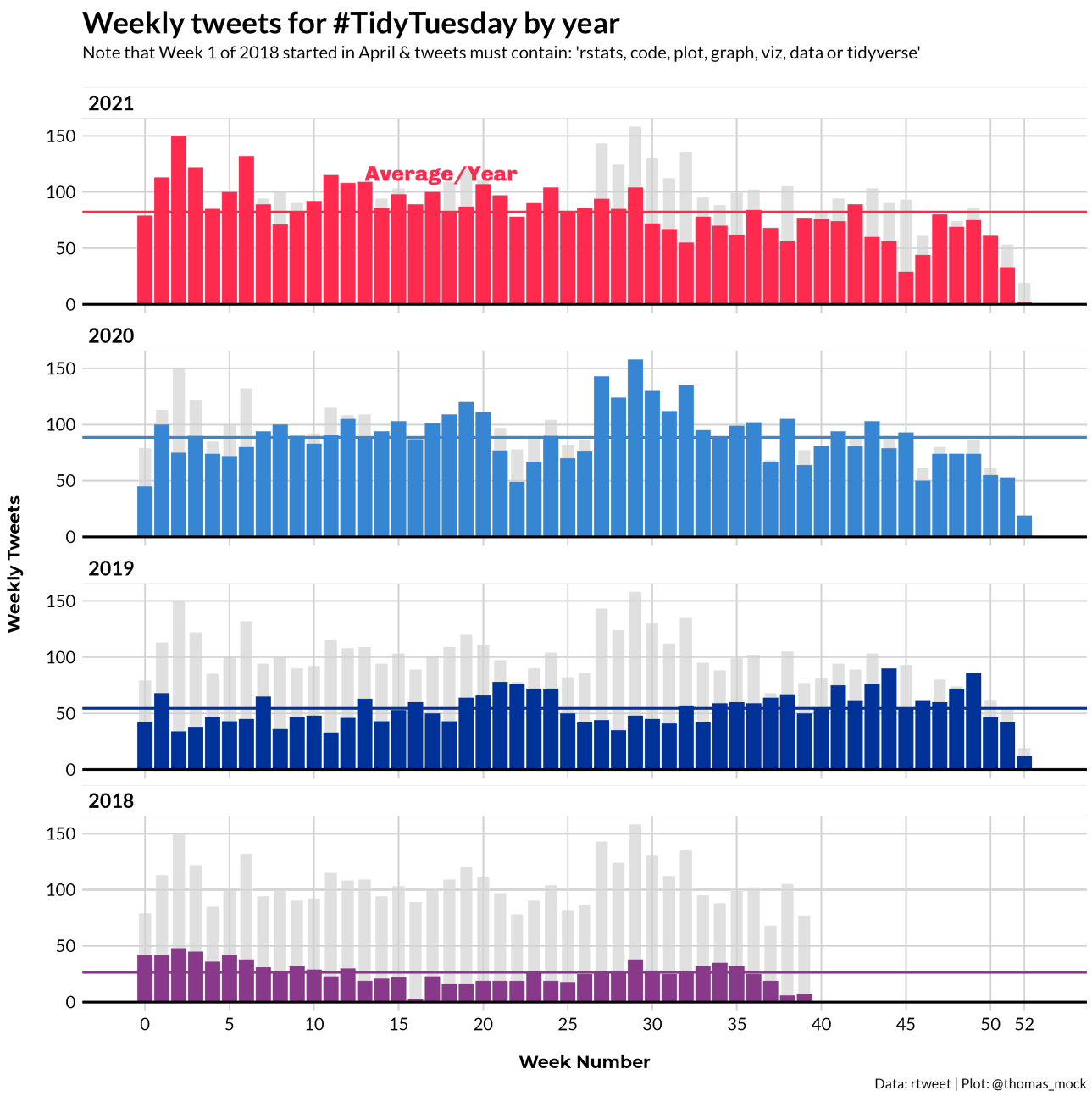
```
col_label_yr <-        tibble  (
  week =         13       ,
  n =         110      ,
  text =          "Average/Year",
  colour =          "#FF2B4F",
  year =         factor  (        2021     )
)

tt_col    <-         sum_df    %>%
  group_by(         week     )         %>%
  mutate  (         max =          max      (         n         )
  ungroup (         )         %>%
  ggplot  (
    aes      (
      x =          week     ,
      y =          n        ,
      fill =          colour   ,
      group =          year
    )         ,
    color =          "white"  ,
  )         +
  geom_col(         aes      (         x =          week     , y =
  geom_col(         )         +
  geom_text(
    data =          col_label_yr, hjust =          0         , vju
    fontface =          "bold"   , family =          "Chivo"  ,
    aes      (         label =          text     , x =          week
    )         +
  facet_wrap(         ~        year     , ncol =          1
  geom_hline(         yintercept =          0         , size =
  geom_hline(
    data =          group_by(         sum_df   , year     )
    aes      (         yintercept =          mean     )         ,
    color =          c        (         "#FF2B4F", "#3686d3", "#0

    size =          1
```

The MockUp

```
    limits = c(-1, 53
  ) +
  scale_color_identity(aesthetics = c(
  tomtom::theme_538() +
  theme(
    legend.position = c(0.1, 0.8
    legend.background = element_blank()
    strip.background = element_rect(fill =
    strip.text = element_text(color = "
    legend.title = element_blank(),
    legend.text = element_text(size = 1
    plot.background = element_blank(),
    plot.title = element_text(size = 24
  ) +
  labs(
    x = "\nWeek Number",
    y = "Weekly Tweets\n",
    caption = "Data: rtweet | Plot: @thomas_mock",
    title = "Weekly tweets for #TidyTuesday by year",
    subtitle = "Note that Week 1 of 2018 started in Ap
  )

ggsave(
  "tt_columns.png",
  plot = tt_col,
  device = ragg::agg_png(
    width = 10,
    height = 10,
    units = "in",
    scaling = 0.8,
    res = 500
  )
)
```

```
Saving 12.5 x 12.5 in image
```

The MockUp



A bar chart of TidyTuesday-tagged tweets across the past 3 years, split into small multiples by year, where each year shows higher weekly peaks over the same week as last year. 2021 is on pace to have the highest number of tweets for most weeks.

Data sourced from the TidyTuesday repo.

Below is a corresponding table telling part of the same story, namely that 2021 has many of the highest weekly tweet counts seen over the past 3 years.

The MockUp

```
  group_by(        week     )        %>%
  filter  (        n        ==       max     (       n
  ungroup (        )        %>%
  arrange (        week     )        %>%
  select  (        year     :        roll_n  )        %>%
  gt      (        )        %>%
  tab_style(
    style =        cell_text(       color =        "#FF2B4F"
    locations =        cells_body(       vars    (       yea
  )        %>%
  tab_style(
    style =        cell_text(       color =        "#3686d3"
    locations =        cells_body(       vars    (       yea
  )        %>%
  fmt_number(       columns =        vars    (       roll_n
  cols_label(
    week =        html    (       "Week<br>Number")       ,
    n =        html    (       "Weekly<br>Tweets")       ,
    roll_n =        html    (       "Rolling<br>Total Tweets
    )        %>%
  tab_header(
    title =        md      (       "**Peak number of tweets
    subtitle =       "2021 has most of the top counts for ea
  )        %>%
  espnscrapeR::      gt_theme_538(       )        %>%
  tab_source_note(
    html    (       "<strong>Data Source: </strong><a href='h
  )
```

```
Warning: `columns = vars(...)` has been deprecated in gt
0.3.0:
* please use `columns = c(...)` instead

Warning: `columns = vars(...)` has been deprecated in gt

0.3.0:
```

The MockUp

```
0.3.0:
* please use `columns = c(...)` instead

Warning: `columns = vars(...)` has been deprecated in gt
0.3.0:
* please use `columns = c(...)` instead
```

Peak number of tweets by week

2021 has most of the top counts for each week

YEAR	WEEK NUMBER	WEEKLY TWEETS	ROLLING TOTAL TWEETS
2021	0	79	79
2021	1	113	192
2021	2	150	342
2021	3	122	464
2021	4	85	549
2021	5	100	649
2021	6	132	781
2020	7	94	630
2020	8	100	730
2020	9	90	820
2021	10	92	1,116
2021	11	115	1,231
2021	12	108	1,339

Data Source: TidyTuesday Repo

Peak users

The last chart is the number of unique contributors by year, regardless of whether they contributed in previous years.

The MockUp

```
  filter  (       str_detect(       tolower (       text
  mutate  (       created_date =       lubridate::      as_
  mutate  (
    year =       year    (       created_date)       ,
    week =       week    (       created_date)       ,
    week =       if_else (       year     >       2018
    year =       factor  (       year    , levels =
  )       %>%
  filter  (       year     %in%     2018    :       2021
  count   (       screen_name, week    , year    )       %>
  mutate  (       n =       1       )       %>%
  arrange (       year    , week    )       %>%
  distinct(       screen_name, year    , .keep_all =
  group_by(       year    )       %>%
  mutate  (       week_roll_n =       cumsum  (       n
  mutate  (
    colour =       case_when(
      year     ==       2021     ~       "#FF2B4F",
      year     ==       2020     ~       "#3686d3",
      year     ==       2019     ~       "#003399",
      year     ==       2018     ~       "#88398a",
      TRUE     ~       "gray80"
    )
  )       %>%
  ungroup (       )       %>%
  mutate  (       roll_n =       cumsum  (       n       )

top_user_sum <-       user_year_sum %>%
  group_by(       year    )       %>%
  arrange (       desc    (       week_roll_n)       )
  group_by(       year    , week    )       %>%

  mutate  (       sum =       sum     (       n       )
  group_by(       year    )       %>%
  mutate  (       mean =       mean    (       sum     )
  slice   (       1       )       %>%
  ungroup (       )
```

The MockUp

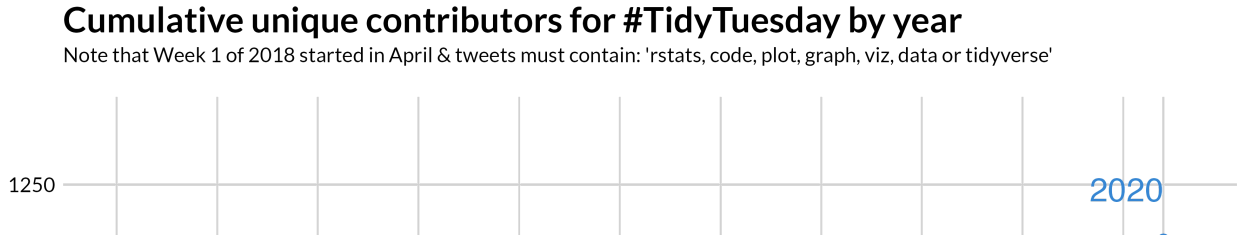
```
geom_step(        size =         1        )         +
geom_point(
  data =         top_user_sum,
  aes      (         col =          colour    )         ,
  size =         2.5      ,
  stroke =          1
)         +
geom_text(
  data =         top_user_sum,
  aes      (         label =          year     )         ,
  size =         8        ,
  hjust =          c        (        1        , 1         , 1
  nudge_y =          50       ,
  vjust =         0
)         +
geom_hline(         yintercept =         0         , size =
scale_y_continuous(
  breaks =         seq      (         0         , 1250     , by =
  limits =         c        (         0         , 1300     )
)         +
scale_x_continuous(
  breaks =         c        (         seq      (         0
  limits =         c        (         0         , 53       )
)         +
scale_color_identity(         aesthetics =         c        (
labs     (
  x =          "\nWeek Number",
  y =          "Unique Users\n",
  caption =          "Data: rtweet | Plot: @thomas_mock",
  title =         "Cumulative unique contributors for #TidyT
  subtitle =          "Note that Week 1 of 2018 started in Ap

)         +
tomtom   ::       theme_538(         )         +
theme    (
  legend.position =         c        (         0.1       , 0.8
  legend.background =          element_blank(         )
```

The MockUp

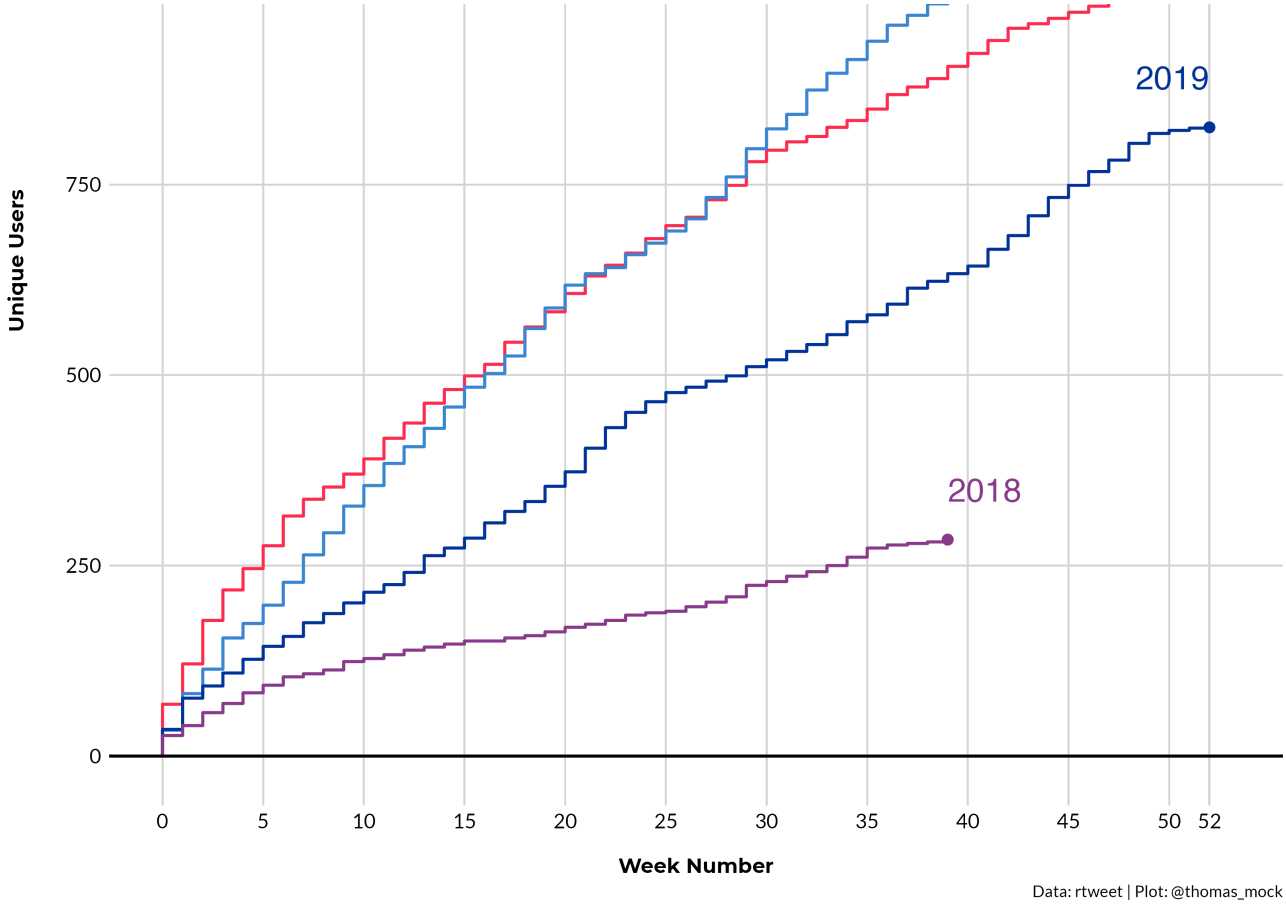
```
    plot.title =        element_text(        size =        24
  )

ggsave  (
  "tt_users.png",
  user_unique_plot,
  device =        ragg    ::      agg_png (
    width =        10       ,
    height =        10       ,
    units =        "in"     ,
    scaling =         0.8      ,
    res =        500
  )
)
```

```
Saving 12.5 x 12.5 in image
```

```
knitr   ::      include_graphics(       "tt_users.png"
)
```



The MockUp



A line chart of unique contributors to TidyTuesday across the past 3 years, where there is large year over year growth. 2021 is on pace to exceed the total unique users from 2020.

Data sourced from the TidyTuesday repo.

Below is a corresponding table telling part of the same story, namely that 2021 has many of the highest weekly user counts recorded over the past 3 years.

```
top_user_sum %>%
  select   (        year     , week     , mean     , roll_n =
  gt       (        )        %>%
  fmt_number(         columns =         vars     (        roll_n
  cols_label(
    week =          html     (        "Unique<br>Weeks")        ,
    mean =          html     (        "Average<br>Weekly Users")
    roll_n =          html     (         "Rolling<br>User Count")
```

The MockUp

```
    subtitle =          "2021 has the highest unique user per w
  )         %>%
  espnscrapeR::       gt_theme_538(         )         %>%
  tab_source_note(
    html     (         "<strong>Data Source: </strong><a href='h
  )
```

```
Warning: `columns = vars(...)` has been deprecated in gt
0.3.0:
* please use `columns = c(...)` instead

Warning: `columns = vars(...)` has been deprecated in gt
0.3.0:
* please use `columns = c(...)` instead
```

Summary of TidyTuesday Contributors by year

2021 has the highest unique user per week

YEAR	UNIQUE WEEKS	AVERAGE WEEKLY USERS	ROLLING USER COUNT
2021	52	28	1,048
2020	52	26	1,178
2019	52	19	825
2018	39	11	284

Data Source: TidyTuesday Repo

Future

Again, I'm ecstatic about how many contributors and unique tweets there have been to TidyTuesday over the past 3 years, but I hope that we as a community can improve the user experience for potential low vision or blind contributors as well.

The MockUp

community.

Expand for Session Info

The MockUp