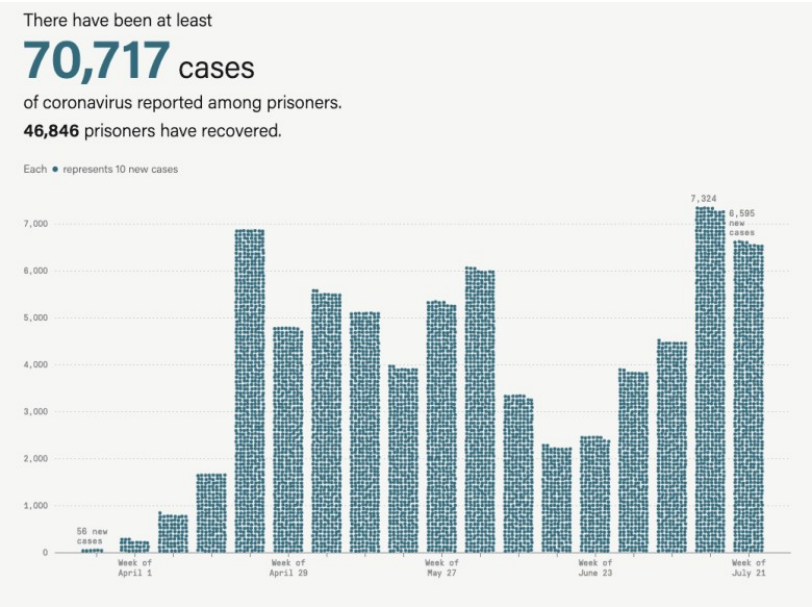
The Marshall Project has a solid story and set of visualizations on the impact of COVID-19 in U.S. prisons. They keep the data (and vis) regularly updated. They do great work and this is an important topic, but this visualization breaks my “ordered grid” OCD:



To be fair, it’s not supposed to line up as the dots are part of an animation process that has them drop from top to bottom and appears to be designed to have an “organic” feel.

We can use the {waffle} package to iron out these wrinkled non-grids into some semblance of order, and try to replicate the chart as much as possible along the way.

Getting the Data

We first need the data and, thankfully, the MP folks provided it…just not in a way you’d expect (or that’s straightforward to use).

Do a “view source” on that URL in your browser and scroll down to line ~1,455 and you should see this:

That’s the data, right on the page, encoded in javascript🤔. This makes sense as it is fueling a javascript visualization and many sites are embedding data right on the page vs fetch via an XHR request to make it easier for web archives to store and retrieve working visualizations. We can totally work with this data, and we’ll do that now, along with getting some boilerplate out of the way:

```
library(V8)         # work with javascript data
library(stringi)    # string ops
library(rvest)      # web scrape
library(ggtext)     # pretty ggplot text with markdown
library(waffle)     # waffle charts // install_github("hrbrmstr/waffle")
library(hrbrthemes) # install_github("hrbrmstr/hrbrthemes") or don't use the
font theme and pick another one
library(tidyverse)  # duh

gg <- glue::glue # for plot labels (later)

# get the page source
pg <- read_html("https://www.themarshallproject.org/2020/05/01/a-state-by-state-look-at-coronavirus-
in-prisons")

# setup a V8 VM context
ctx <- v8()

# grab the "data" and make it a V8 VM object
html_nodes(pg, xpath=".//script[contains(., 'var STATES_DATA')]") %>%
  html_text() %>%
  ctx$eval()

# get the data into R
states_data <- ctx$get("STATES_DATA")

glimpse(states_data)
## Rows: 918
## Columns: 20
## $ ``                  "1", "2", "3", "4", "5", "6", "7", "8", "9", "10",
"11", "12", "13", "14", "15", "16", "17", "18", "19", "20",…
## $ name                "Alabama", "Alaska", "Arizona", "Arkansas",
"California", "Colorado", "Connecticut", "Delaware", "Florida", "G…
## $ abbreviation        "AL", "AK", "AZ", "AR", "CA", "CO", "CT", "DE", "FL",
"GA", "HI", "ID", "IL", "IN", "IA", "KS", "KY", "LA", "M…
## $ week_of             "2020-03-26", "2020-03-26", "2020-03-26", "2020-03-26",
"2020-03-26", "2020-03-26", "2020-03-26", "2020-03-26"…
## $ unrevised_cases     "NA", "NA", "NA", "NA", "NA", "NA", "NA", "NA", "NA",
"NA", "NA", "NA", "NA", "NA", "NA", "NA", "NA", "NA", "N…
## $ cases               "0", "0", "0", "0", "1", "0", "NA", "0", "0", "4", "0",
"0", "3", "0", "0", "0", "NA", "0", "0", "0", "9", "23…
## $ new_cases           "0", "0", "0", "0", "1", "0", "NA", "0", "0", "4", "0",
"0", "3", "0", "0", "0", "NA", "0", "0", "0", "9", "23…
## $ filled_cases        "0", "0", "0", "0", "1", "0", "0", "0", "0", "4", "0",
"0", "3", "0", "0", "0", "0", "0", "0", "0", "9", "23",…
## $ case_rate           "0", "0", "0", "0", "0.0812809883768187", "0", "NA",
"0", "0", "0.728318857996031", "0", "0", "0.7865757734661…
## $ deaths              "0", "0", "0", "0", "0", "0", "0", "0", "0", "1", "0",
"0", "0", "0", "0", "0", "0", "NA", "0", "0", "0", "0", "0",…
## $ new_deaths          "0", "0", "0", "0", "0", "0", "0", "0", "0", "1", "0",
"0", "0", "0", "0", "0", "0", "NA", "0", "0", "0", "0", "0",…
## $ filled_deaths       "0", "0", "0", "0", "0", "0", "0", "0", "0", "1", "0",
"0", "0", "0", "0", "0", "0", "0", "0", "0", "0", "0", "0", …
## $ death_rate          "0", "0", "0", "0", "0", "0", "0", "0", "0",
"0.182079714499008", "0", "0", "0", "0", "0", "0", "NA", "0", "0"…
## $ staff_multiples     "NA", "NA", "NA", "NA", "NA", "NA", "NA", "NA", "NA",
"NA", "NA", "NA", "NA", "NA", "NA", "NA", "NA", "NA", "N…
```

```
## $ prisoner_multiples  "NA", "NA", "NA", "NA", "NA", "NA", "NA", "NA", "NA",
"NA", "NA", "NA", "NA", "NA", "NA", "NA", "NA", "NA", "N…
## $ tested              "NA", "4", "15", "0", "166", "NA", "NA", "4", "NA",
"NA", "NA", "10", "13", "NA", "NA", "0", "NA", "32", "NA",…
## $ as_of_date          "NA", "2019-12-31", "2020-04-15", "2020-02-29",
"2020-04-15", "NA", "NA", "2020-04-15", "NA", "NA", "NA", "202…
## $ march_pop           "NA", "4997", "42282", "18181", "123030", "NA", "NA",
"5042", "NA", "NA", "NA", "7816", "38140", "NA", "NA", "…
## $ april_pop           "NA", "4997", "41674", "18181", "118466", "NA", "NA",
"4624", "NA", "NA", "NA", "7641", "36904", "NA", "NA", "…
## $ test_rate           "NA", "8.00480288172904", "3.54760891159359", "0",
"13.4926440705519", "NA", "NA", "7.93335977786593", "NA", "…
```

The comments in the code go a long way, but jist is that we extract out the javascript block containing that `var STATES_DATA…` data, have {V8} wrangle it in javascript for us, then get the result and take a look at it. Now for the real work.

Making the Data Useful

We need — at a minimum — dates and numbers. We're also going to mimic the visualization, so we'll be dividing new case counts by 10 for the "1 dot == 10 cases" waffle chart and creating useful axis labels. This is pretty basic wrangling:

```
states_data %>%
  select(week_of, new_cases) %>%
  mutate(
    week_of = as.Date(week_of),
    new_cases = suppressWarnings(as.numeric(new_cases))
  ) %>%
  count(week_of, wt = new_cases) %>%
  arrange(week_of) %>%
  mutate(
    wk = format(week_of, "Week of\n%b %d"),
    div10 = as.integer(round(n/10)),
  ) %>%
  as_tibble() -> cases

glimpse(cases)
## Rows: 18
## Columns: 4
## $ week_of  2020-03-26, 2020-04-01, 2020-04-08, 2020-04-15, 2020-04-22,
2020-04-29, 2020-05-06, 2020-05-13, 2020-05-20, 2020-05-27, …
## $ n        56, 268, 810, 1672, 6872, 4788, 5538, 5115, 3940, 5323, 6027,
3335, 2258, 2452, 3856, 4488, 7324, 6595
## $ wk       "Week of\nMar 26", "Week of\nApr 01", "Week of\nApr 08", "Week
of\nApr 15", "Week of\nApr 22", "Week of\nApr 29", "Week o…
## $ div10    6, 27, 81, 167, 687, 479, 554, 512, 394, 532, 603, 334, 226, 245,
386, 449, 732, 660
```

Using the {waffle} package to make "waffle bar charts" means we'll end up with panels/strips which will become "axis labels". I like the fact that the MP folks did not label each week, so we'll have to account for that as well. One of the simplest ways to do that is to make those labels spaces, but a unique number of them since we're going to make an ordered factor to ensure the strips are in the right order. This is also pretty straightforward:

```
cases$wk[c(1, 3:5, 7:9, 11:13, 15:17)] <- stri_pad("", 1:13)
cases$wk <- fct_inorder(cases$wk)
```

The vector of numbers in the first line are the weeks we want to be blank and we'll turn them into space-

padded strings, each with an increasing number of spaces, then we'll turn the entire vector of weeks into a factor in the right order.

Making the Chart

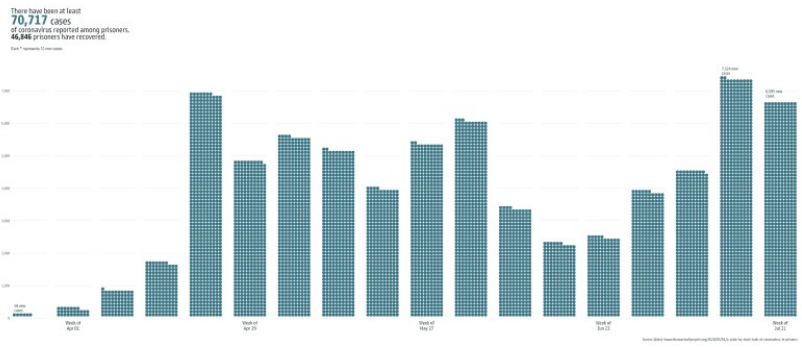
The rest is all {ggplot2} magic, so let's get the whole plot code out of the way before talking about it:

```
ggplot() +
  geom_waffle(
    data = cases,
    aes(fill = "new cases", values = div10),
    flip = TRUE, n_cols = 10, radius = unit(3, "pt"),
    color = "white"
  ) +
  geom_text(
    data = tibble(
      idx = c(1, 17, 18),
      wk = cases$wk[idx],
      y = (cases$div10[idx] %/% 10),
      actual_cases = cases$n[idx],
      lab = gg("{scales::comma(actual_cases, 1)} new\ncases")
    ),
    aes(1, y, label = lab),
    vjust = 0, hjust = 0, nudge_y = 2,
    size = 3.5, family = font_gs, lineheight = 0.875
  ) +
  scale_y_continuous(
    expand = c(0, 0.125),
    breaks = seq(0, 70, 10),
    labels = scales::comma(seq(0, 7000, 1000)),
    limits = c(0, 80)
  ) +
  scale_fill_manual(
    values = c("#366b7b")
  ) +
  facet_wrap(~wk, nrow=1, strip.position = "bottom") +
  coord_fixed() +
  labs(
    x = NULL, y = NULL,
    title = "There have been at least
**70,717** cases
of coronavirus reported among prisoners.
**46,846** prisoners have recovered.",
    subtitle = "Each • represents 10 new cases.",
    caption = "Source: (data) "
  ) +
  theme_ipsum_gs(
    grid="Y",
    strip_text_family = font_gs, strip_text_face = "plain",
    plot_title_family = font_gs, plot_title_face = "plain",
    subtitle_family = font_an, subtitle_face = "plain", subtitle_size = 10
  ) +
  theme(
    legend.position = "none",
    strip.text = element_text(hjust = 0.5),
    axis.text.x = element_blank(),
    panel.spacing.x = unit(20, "pt"),
```

```
    plot.title = element_markdown(),
    plot.subtitle = element_markdown(),
  )
```

There's quite a bit going on there, so let's break it down:

- We're telling `geom_waffle()` to use our data, and giving it a single category to fill (as there is only one) along with the number of elements in the category. The `radius` parameter lets us have non-square "dots", and `n_cols` + `flip` sets up the grid to match the one from MP.
- We need labels on top, too (just three of them) so we'll pick the vector indices of the ones with labels and get the week strip labels, y positions, new case counts for that day, and an appropriately formatted label and plot them. We're starting the label at the first X position in each strip and plotting the labels at the height of the "bar".
- We're customizing the Y scale to reflect the 1 == 10 representation of the data and using the same blue as MP did for the fill scale.
- To get them all to mimic a real X axis, we're ensuring there's only one row of facets and putting the facet labels at the bottom.
- By using `coord_fixed` we can get circles (or as close to them as you like)
- We're using some markdown in the `labs()`, courtesy of {ggtext}'s `element_markdown()` and setting some font stylings in the base theme (use a different one if you get font errors or read the docs). We rely on this to "fake" a legend.
- Finally, we tweak strip positions and some formatting to produce:



(You likely need to view that in your own plot window in R/RStudio or zoom in a bit)

FIN

If you spend some more time on it you can get *super-close* to the Marshall Project's finished product.

A bonus from scraping is that you also get two more datasets from the page: `STATES_DATA` and `STATE_NOTES`:

```
glimpse(ctx$get("STATES_RATES"))
## Rows: 51
## Columns: 23
## $ ``                  "1", "2", "3", "4", "5", "6", "7", "8", "9", "10",
"11", "12", "13", "14", "15", "16", "17", "18", "19", "20",…
## $ name                "Alabama", "Alaska", "Arizona", "Arkansas",
"California", "Colorado", "Connecticut", "Delaware", "Florida", "G…
## $ abbreviation        "AL", "AK", "AZ", "AR", "CA", "CO", "CT", "DE", "FL",
"GA", "HI", "ID", "IL", "IN", "IA", "KS", "KY", "LA", "M…
## $ unrevised_cases     NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA,
NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA…
## $ cases               "165", "16", "684", "3789", "7066", "783", "1344",
"508", "3898", "1113", "0", "763", "343", "729", "333", "91…
## $ new_cases           "27", "14", "115", "307", "608", "115", "1", "213",
"1266", "49", "0", "69", "6", "1", "109", "1", "146", "50"…
## $ filled_cases        "165", "16", "684", "3789", "7066", "783", "1344",
```

```
"508", "3898", "1113", "0", "763", "343", "729", "333", "91…
## $ case_rate           "77.9994327313983", "32.0192115269162",
"164.131112924125", "2084.04378197019", "596.458055475833", "449.68986…
## $ deaths              "14", "0", "13", "25", "40", "3", "7", "7", "34", "26",
"0", "0", "13", "20", "1", "4", "6", "16", "0", "8", "…
## $ new_deaths          "2", "0", "0", "9", "5", "0", "0", "0", "5", "1", "0",
"0", "0", "0", "0", "0", "2", "0", "0", "0", "0", "0", …
## $ filled_deaths       "14", "0", "13", "25", "40", "3", "7", "7", "34", "26",
"0", "0", "13", "20", "1", "4", "6", "16", "0", "8", "…
## $ death_rate          "6.61813368630046", "0", "3.11945097662811",
"13.750618777845", "3.376496209883", "1.72294968986906", "5.72925…
## $ staff_multiples     "NA", "NA", "NA", "NA", "NA", "NA", "NA", "NA", "NA",
"NA", "NA", "NA", "NA", "NA", "NA", "NA", "NA", "NA", "NA", "N…
## $ prisoner_multiples  "NA", "NA", "NA", "NA", "NA", "13723", "NA", "NA",
"NA", "NA", "NA", "1884", "NA", "NA", "NA", "NA", "NA", "NA…
## $ as_of_date          "2020-01-31", "NA", "2020-04-15", "2020-02-29",
"2020-04-15", "2020-03-31", "2020-04-01", "NA", "NA", "NA", "2…
## $ march_pop           "21154", "NA", "42282", "18181", "123030", "17600",
"12422", "NA", "NA", "NA", "4631", "7816", "38140", "26891…
## $ april_pop           "21154", "NA", "41674", "18181", "118466", "17412",
"12218", "NA", "NA", "NA", "4631", "7641", "36904", "26891…
## $ test_rate           "313.888626264536", "NA", "1176.99284925853",
"4720.86243880975", "5048.53713301707", "4241.90213645762", "819…
## $ recovered           "41", "2", "376", "2970", "4940", "628", "1324", "391",
"NA", "881", "0", "100", "307", "716", "208", "906", "…
## $ date                "20200721", "20200721", "20200721", "20200721",
"20200721", "20200721", "20200721", "20200721", "20200721", "2…
## $ case_ratio          "-45.6429050602488", "-7.23368674670148",
"-19.650267894127", "1714.81334912848", "488.054058525172", "538.378…
## $ death_ratio         "149.040167449448", "-100", "-22.1877969354022",
"1009.53669385711", "72.0346501657667", "-38.5633906796423", …
## $ test_ratio          "-74.3653779188402", "NA", "6.07104744754024",
"224.056036286688", "205.156799892597", "441.281822953195", "34…

glimpse(ctx$get("STATE_NOTES"))
## Rows: 18
## Columns: 3
## $ state  "TN", "TN", "VA", "NM", "NM", "MN", "MN", "VT", "RI", "RI", "MI",
"MD", "CT", "AK", "DE", "HI", "LA", "MA"
## $ type   "prisoners", "staff", "staff", "prisoners", "staff", "prisoners",
"staff", "prisoners", "prisoners", "staff", "prisoners", …
## $ text   "After testing everyone in all of their prisons, Tennessee has said
it is releasing the total number of tests conducted and……
```