

USING R TO REDUCE TECHNICAL DEBT

Children's Hospital of Philadelphia
Jake Riley

12/8/2021



Jake Riley

- Clinical Analyst @ Children's Hospital of Philadelphia
- Author of several R packages



- Learned most of this from time at CHOP

rocqi

- R for OCQI



- data connections

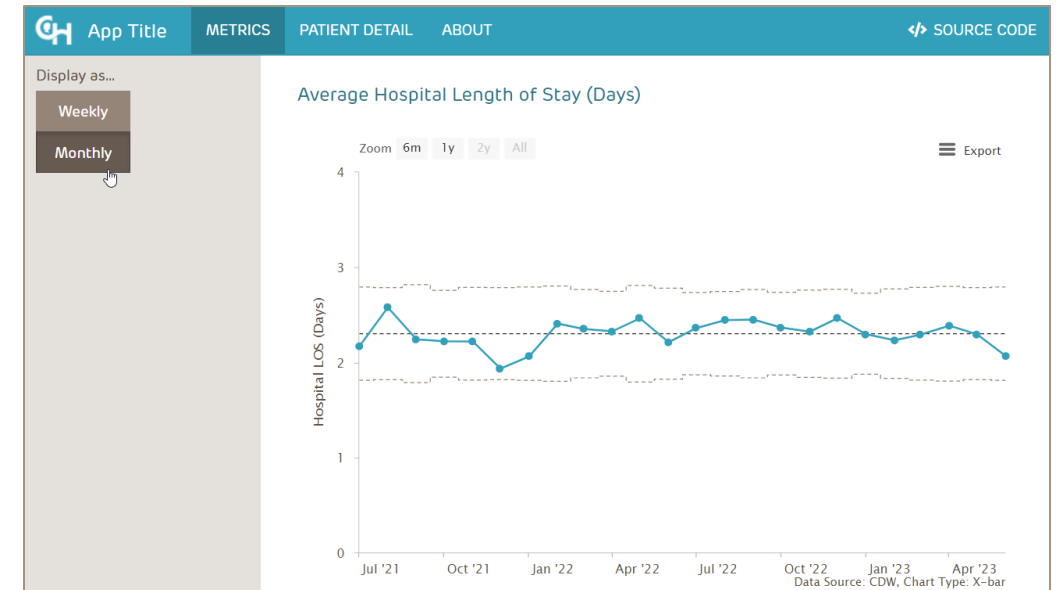
```
run_sql("select pat_key from patient limit 10")
run_sql_file("fact_ortho_patients.sql")
```

- visualization

- Rmd templates

- data sets

blue-1	brown-1 rules	green-1	pink-1
blue-2	brown-2	green-2	pink-2
blue-3	brown-3	green-3	pink-3
blue-4	brown-4 footer-bg	green-4	pink-4
blue-5 blue primary	brown-5 text	green-5 green	pink-5 pink
blue-6 primary-dark	brown-6	green-6	pink-6
blue-7	brown-7	green-7	pink-7
blue8	brown8	green8	pink8



technical debt

“is a concept in software development that reflects the **implied cost of additional rework** caused by choosing an easy (limited) solution now instead of using a better approach that might take longer” - wikipedia

- copy pasting code instead of using functions
- hard coding things that could be soft coded
- hard to read:
 - sparse annotation
 - messy style (lint)
 - object names too abstract (a, a1, a2, ...)

charts got us started

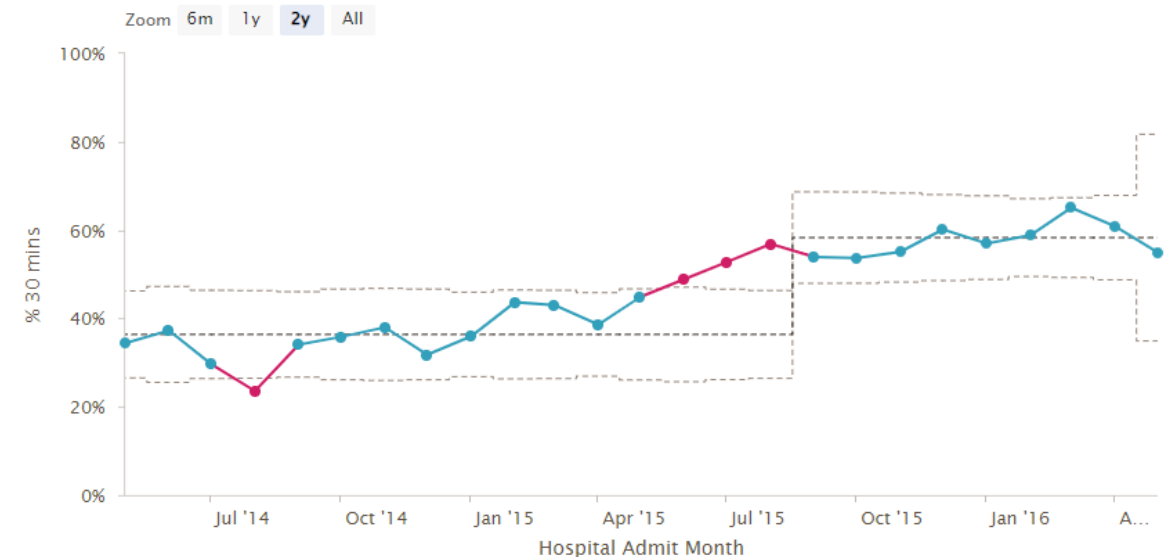
- `spc()` & `hc_spc()` – uses `qicharts2`, `ggplot2`, `highcharter`

```
highchart(type = "stock") ▷  
  hc_add_series(  
    df, name = chart_title, hcaes(x = x, y = y),  
    type = "line", dashStyle = "Solid", color = chop_colors("blue"),  
    marker = marker_point(chop_colors("blue"))  
  ) ▷  
  hc_add_series(  
    df, name = "UCL", hcaes(x = x, y = ucl),  
    type = "line", dashStyle = "ShortDash", color = chop_colors("brown2"),  
    step = "center", marker = list(enabled = FALSE)  
  ) ▷  
  hc_add_series(  
    df, name = "Centerline", hcaes(x = x, y = cl),  
    type = "line", dashStyle = "ShortDash", color = chop_colors("blue"),  
    step = "center", marker = list(enabled = FALSE)  
  ) ▷  
  hc_add_series(  
    df, name = "LCL", hcaes(x = x, y = lcl),  
    type = "line", dashStyle = "ShortDash", color = chop_colors("brown2"),  
    step = "center", marker = list(enabled = FALSE)  
  ) ▷  
  hc_title(text = "Some title about this chart") ▷  
  hc_subtitle(text = "P chart") ▷  
  hc_credits(enabled = TRUE, text = "Source: CDW") ▷  
  hc_xAxis(title = list(text = "Hospital Admit Date")) ▷  
  hc_yAxis(title = list(text = "% 30 mins")) ▷  
  hc_theme_chop()
```

```
hc_spc(  
  data = df,  
  x = x,  
  chart = "p",  
  title = "Some title about this chart",  
  subtitle = "P chart",  
  caption = "Source: CDW",  
  xlab = "Hospital Admit Date",  
  ylab = "% 30 mins"  
)
```

Sepsis Abx Intervention

P chart



easy queries

- Use ODBC connections: connect, run query, disconnect

```
run_sql(  
  sql = "select * from abc",  
  dsn = "cdwprd",  
  lowercase_names = TRUE  
)
```

- Also have

```
run_sql_file(  
  path = "fact_ortho.sql",  
  dsn = "cdwprd",  
  lowercase_names = TRUE  
)
```

- Returns a tibble

date helpers

- `fiscal_month("2021-07-01")` returns a factor list July -> June

```
fiscal_month("2020-07-01")
#> [1] Jul
#> Levels: Jul Aug Sep Oct Nov Dec Jan Feb Mar Apr May Jun

fiscal_month("2020-11-01", format = "%B")
#> [1] November
#> 12 Levels: July August September October November December January ... June
```

```
ggplot(
  weekend_surgeries,
  aes(x = fiscal_month(surgery_dt), ...)
) +
  geom_line()
```



```
fiscal_year("2021-01-01")
#> [1] 2021

fiscal_year("2021-07-01")
#> [1] 2022
```

```
# Fiscal Quarter can be formatted using 'glue' syntax
fiscal_quarter("2020-07-01")
#> FY'21 Q1

fiscal_quarter("2020-07-01", "{q}")
#> 1

fiscal_quarter("2020-07-01", "Q{q}'{yy}")
#> Q1'21

fiscal_quarter("2020-07-01", "{yyyy} Q{q}")
#> 2021 Q1
```

making pretty tables

- remove **_key** fields
- **_date** fields nicely formatted
- **_ind** fields
 - convert 0/1 values to Y/N (customizable)
 - remove **_ind** suffix
- Columns become title case
 - has wordbank to keep uppercase (customizable)
- naming conventions
 - prefixes: **n_** → #, **pct_** → %
 - suffixes: **_mins** → (Minutes),
 - also **hr**, **wk**, **mon**, **qtr**, **yr**, **fy** and plural **hrs**, **wks**, etc

visit_key	age_years	sex	ed_arrival_date	icd10_code	bone	ed_los_hours	sedation_pathway_ind	outside_imaging_ind	ketamine_given_ind	time_to_ketamine_mins
All	All	/	All	All	All	All	All	All	All	All
35590394	10	M	2021-07-04	S52.5	radius	5.3	1	0	1	263
85458397	14.2	M	2021-07-04	S59.0	ulna	5	0	0	0	
40604796	10.3	F	2021-07-04	S59.2	radius	5.5	1	0	0	
95357307	12.6	M	2021-07-05	S52.6	ulna	6	1	0	1	263
54304343	10.1	M	2021-07-05	S52.6	ulna	6.6	1	0	0	

Age Years	Sex	ED Arrival Date	ICD10 Code	Bone	ED LOS Hours	Sedation Pathway	Outside Imaging	Ketamine Given	Time To Ketamine (Minutes)
All	All	All	All	All	All	All	All	All	All
10	M	07/04/2021	S52.5	radius	5.3	Yes	No	Yes	263
14.2	M	07/04/2021	S59.0	ulna	5	No	No	No	
10.3	F	07/04/2021	S59.2	radius	5.5	Yes	No	No	
12.6	M	07/05/2021	S52.6	ulna	6	Yes	No	Yes	263

CHOP style guide is easy to follow

- `chop_colors("blue-4", "pink2")` – returns hex codes
- `show_chop_colors()` – returns a plot or a table

`show_chop_colors()`

dark-blue	blue-1	brown-1 rules	green-1	pink-1
white	blue-2	brown-2	green-2	pink-2
	blue-3	brown-3	green-3	pink-3
	blue-4	brown-4 footer-bg	green-4	pink-4
	blue-5 blue primary	brown-5 text	green-5 green	pink-5 pink
	blue-6 primary-dark	brown-6	green-6	pink-6
	blue-7	brown-7	green-7	pink-7
	blue-8	brown-8	green-8	pink-8

```
> show_chop_colors("table")
# A tibble: 9 x 7
  shade blue      pink      green      brown `dark-blue` white
  <int> <chr>    <chr>    <chr>    <chr>    <chr>    <chr>
1     1 #d6ecf1 #f6d2e0 #e4ebd8 #e0ddda -      -
2     2 #add9e4 #eca4c1 #c9d7b2 #c2bbb5 -      -
3     3 #85c6d6 #e377a3 #afc28b #a39990 -      -
4     4 #5cb3c9 #d94984 #94ae65 #958579 -      -
5     5 #33a0bb #d01c65 #799a3e #665546 -      -
6     6 #26778b #9c154c #5b732f #4d4035 -      -
7     7 #1a505e #680e33 #3d4d1f #332b23 -      -
8     8 #0d282f #340719 #1e2710 #191512 -      -
9    NA -      -      -      -      #005587 #fbfbfb
```

easy themes

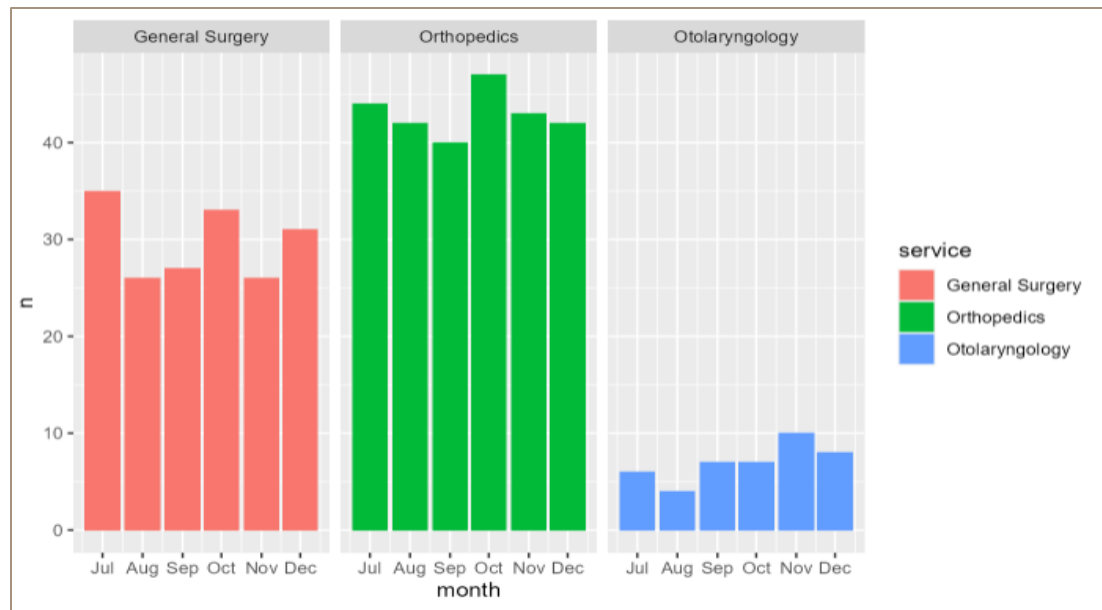
theme_chop() & hc_theme_chop()

- font family, font sizes, legend position, no grid lines, white background

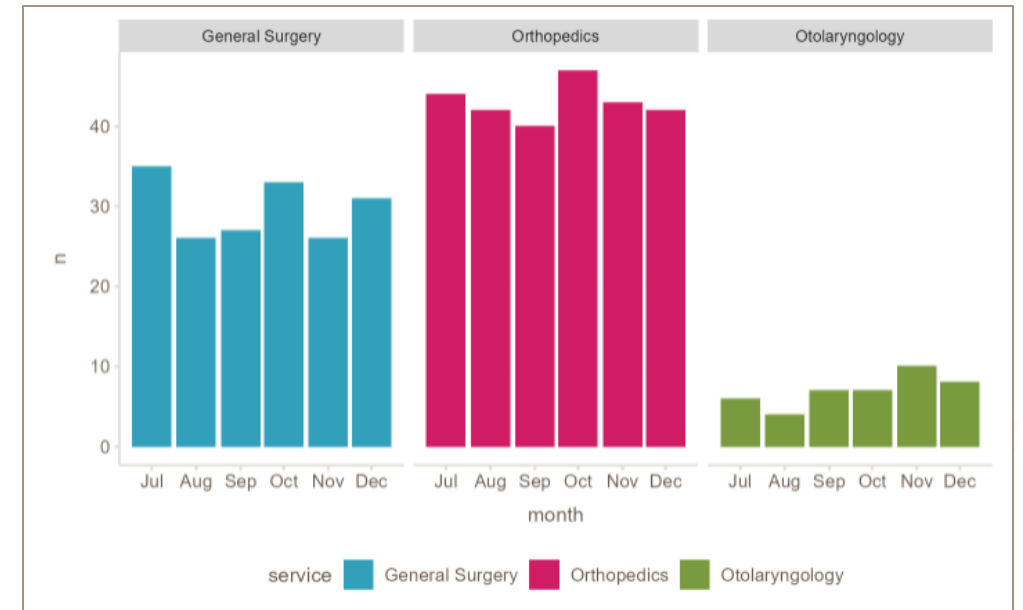
scale_fill_chop() & scale_color_chop()

- cycles through our CHOP colors

original



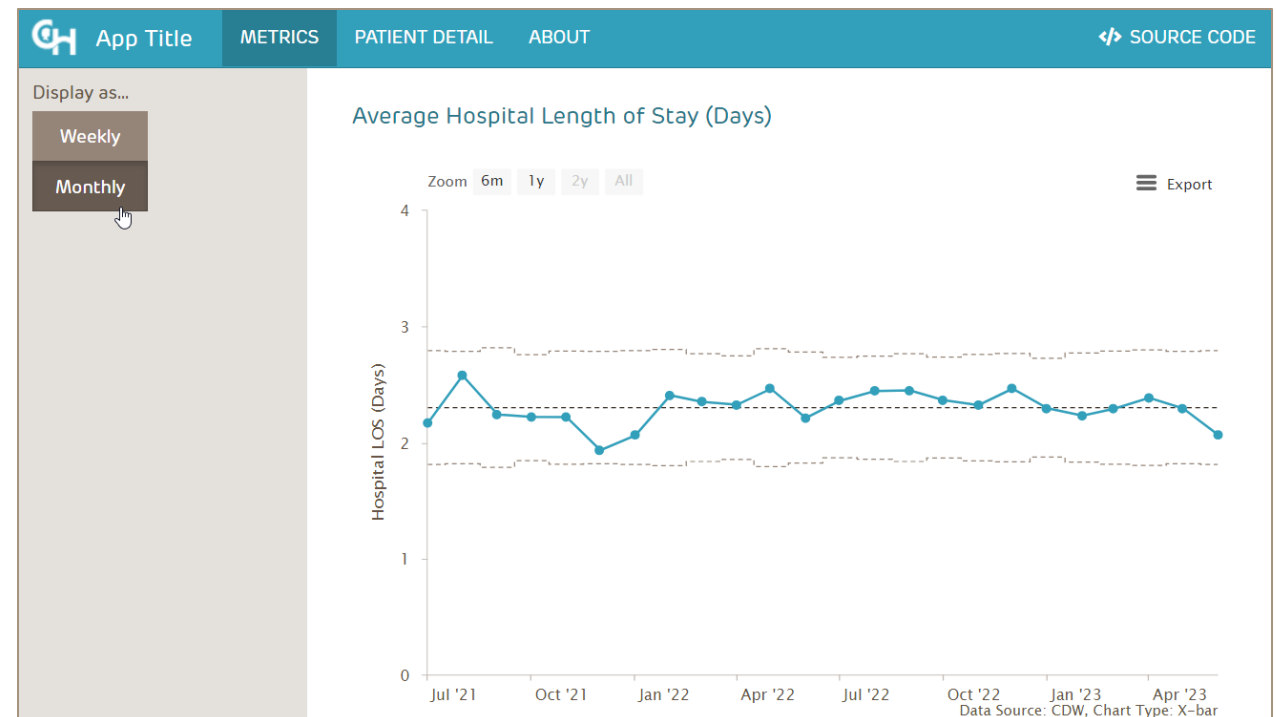
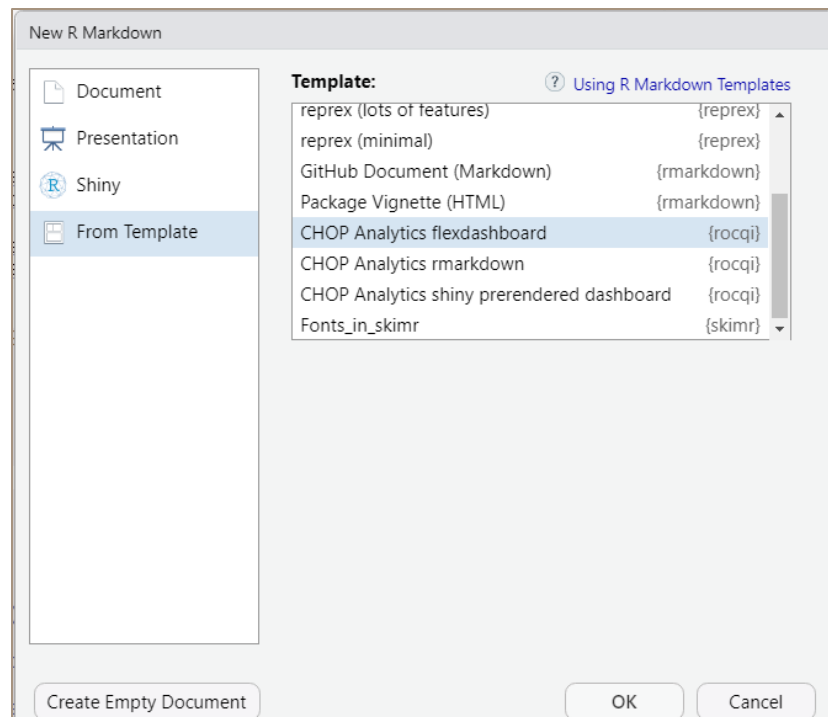
theme_chop() + scale_fill_chop()



ready-to-go templates

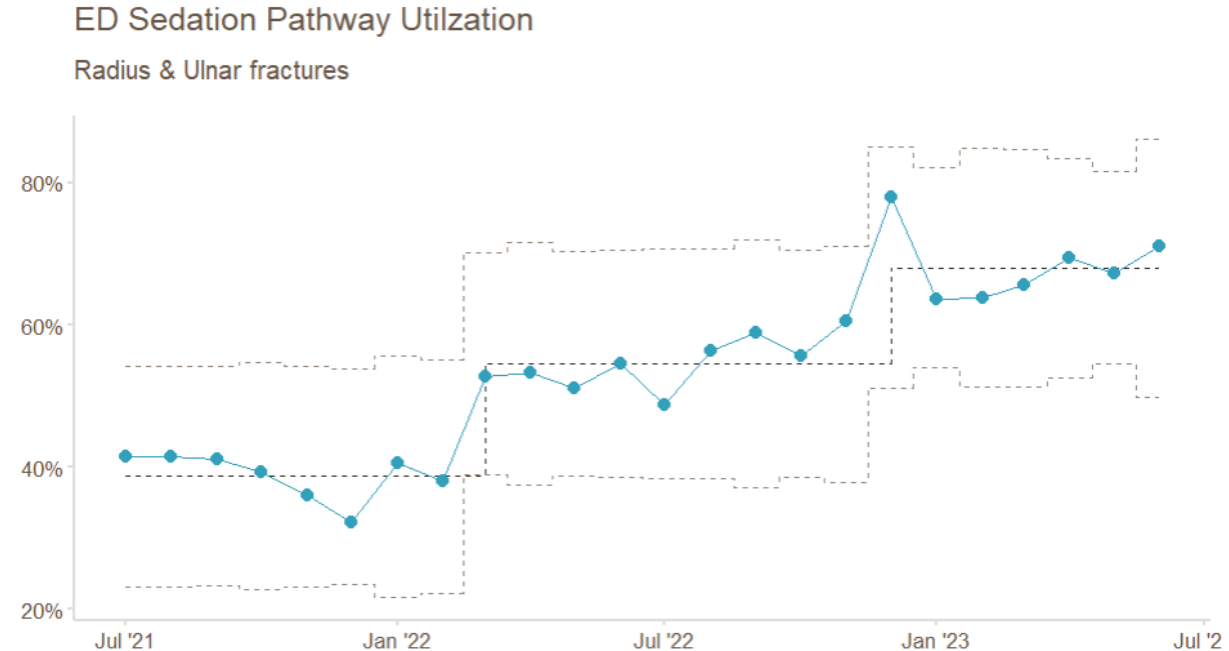
- fonts, colors, logo, standard layout
- 3 templates:
 - flexdashboard
 - rmarkdown
 - shiny prerendered

```
1 ---
2 title: "App Title"
3 runtime: shiny
4 output:
5   flexdashboard::flex_dashboard:
6     orientation: rows
7     vertical_layout: fill
8     source_code: embed
9     logo: https://github.research.chop.edu/pages/CQI/flexdashboard-theme/images/logo/chop-icon-header.png
10    favicon: https://www.chop.edu/sites/all/themes/chop/favicon.ico
11    css:
12      - https://github.research.chop.edu/pages/CQI/chop-bootstrap/bootstrap-3/bootstrap.min.css
13      - https://github.research.chop.edu/pages/CQI/flexdashboard-theme/css/flexdashboard.min.css
14 ---
15
```



built in data sets

- ed_fractures & surgeries
- used in help file documentation, vignettes, & slack for troubleshooting
- reflects the kinds of distributions & problems we often see in the data
- created programmatically, no sensitive data



Source: rocqi

honorable mentions

`nice_display_names()`

- uses regex and wordbank to nicely title a vector of values

`recode_indicator()`

- recodes 0, 1, and NA values

`remove_incomplete_end_dates()`

- removes data if still in current timeframe (day, week, month, year)

`add_date_columns()`

- add columns based on date field `_day`, `_week`, `_month`, etc via `floor_date()`

how it's done

Prioritization

- Monthly sprint meetings – GitHub issues > project board
- we have merged 248 PRs over 33 releases

Maintenance

- feature branches → committed to GitHub (enterprise)
- PR → code review & CI (continuous integration) via webhooks & Jenkins
 - tests – [testthat](#) & [covr](#)
 - style – [tidyverse](#) style guide, [lintr](#)
 - spelling
- main branch regularly updated with periodic releases → CHOPRAN

Documentation

- [roxygen2](#) for all function documentation
- [pkgdown](#) for rendering the package site

impact

- 18 authors, 3 maintainers
 - Adam Rudofker • Brendan Graham • Christian Minich • Connie Tan • Emily Schriver • Ezra Porter • Jake Riley • Joe Mirizio • Kyle Winsor • Matt Devine • Matt Dye • Max Seidman • Nathaniel Schmucker • Paul Wildenhain • Renée Bruhn • Ryan Hawkins • Yuchen Zhang • Zach Dravis
- several of us have since authored our own packages
- contribution back to [highcharter](#), [qicharts2](#), [ggplot2](#) with bugs, feature requests, and pull requests