TBEP-CC

Tampa Bay Estuary Program: Climate Change Indicators

Benjamin D. Best Marcus W. Beck

2024-05-01

Table of contents

Preface

This is a Quarto book.

To learn more about Quarto books visit https://quarto.org/docs/books.

```r
1 + 1
```

```
[1] 2
```

1 Introduction

The Tampa Bay Estuary Program compiles environmental data from multiple partners to report on the status and trends of key indicators of bay health (Monitoring and Indicators Plan). Indicators and reporting methods currently exist for several components of the CCMP, including water quality, benthic and sediment condition, fish communities, seagrasses, and tidal creeks. Open-source methods have been developed using the tbeptools R package (Beck, Schrandt, et al. 2021) that facilitates data synthesis and routine reporting for each of these indicators.

Additional indicators of bay health will complement the existing set of indicators. Key indicators currently missing from TBEP routine reporting products include those related to climate change. The Tampa Bay Regional Planning Council has produced the Regional Resilience Action Plan (RRAP) to assist local governments and municipalities in planning resilience activities in response to climate change. Fundamental to these activities is the identification of robust indicators of the local effects of climate change. These may include data descriptive of sea-level rise, droughts, heat waves, or storm frequency/intensity. The TBEP is supportive of these efforts and intends to identify several climate change indicators that can be operationalized for routine reporting, either through conventional summary graphics or more interactive web-based platforms.

This project will identify appropriate climate change indicators to support the TBEP and its partners in making informed planning decisions. Although the primary goal of TBEP is the management of bay health, these indicators could also be used to support community resilience planning as described in the RRAP. Appropriate data sources will be identified, with emphasis on those that are stable and planned to be reliable sources in the future. Identifying indicators that represent relative risk of climate change impacts will be emphasized, as opposed to indicators that simply measure change. All project activities will leverage open science principles as described in the TBEP Strategic Plan (Burke and Amaral 2020) and Data Management SOP (Beck, Raulerson, et al. 2021).

2 Data

2.1 Task 1. Assessment of available data and coverage

Data descriptive of the risks of climate change can be obtained from several sources. These may include weather or climatological data, long-term tidal gauge data, or in situ water measurements responsive to climate change. Weather and climatological data could be obtained from local weather stations with long-term data, e.g., Tampa International Airport, and could include measures of air temperature, precipitation, and/or storm intensity/frequency. Tidal gauge data are readily available from the NOAA PORTS data retrieval system. Lastly, in situ water measurements could include water temperature, changes in flow hydrology, salinity, and/or pH. Data used to evaluate potential risks related to ocean acidification should also be explored.

The permanency and ease of access of each data source should be noted when making recommendations on indicators to operationalize. Further, indicators that communicate the risks associated with climate change are preferred, as opposed to those that simply indicate change. An example is the number of days in a year when temperature exceeds a critical threshold, as compared to temperature alone. An additional example is frequency of sunny day flooding events, as compared to tidal gauge measurements alone.

```r
if (!"librarian" %in% rownames(installed.packages()))
  install.packages("librarian")
librarian::shelf(
  dplyr, dygraphs, glue, here, leaflet, lubridate, sf,
  tbep-tech/tbeptools,
  RColorBrewer, readr, rnoaa, terra, tidyr, webshot2,
  quiet = T)

# explicitly list packages for renv::dependencies(); renv::snapshot()
library(dplyr)
library(dygraphs)
library(glue)
library(here)
library(leaflet)
library(librarian)
```

```r
library(lubridate)
library(RColorBrewer)
library(readr)
library(rnoaa)
library(sf)
library(tbeptools)
library(terra)
library(tidyr)
library(webshot2)

options(readr.show_col_types = F)
```

2.2 Temperature

2.2.1 Observed

The `rnoaa` R package uses NOAA NCDC API v2, which only goes to 2022-09-15.

- NCEI Web Services | Climate Data Online (CDO) | National Center for Environmental Information (NCEI)
- Data Tools | Climate Data Online (CDO) | National Climatic Data Center (NCDC)

2.2.1.1 Weather stations

- Tampa International Airport
  - Start Date: 1939-02-01
  - End Date: today - 3 days

Got token at ncdc.noaa.gov/cdo-web/token. Added variable `NOAA_NCDC_CDO_token` to:

- locally:

  ```r
  file.edit("~/.Renviron")
  ```

- on GitHub: Repository secrets in Actions secrets · tbep-tech/climate-change-indicators
- GCHN readme
  - `PRCP`: Precipitation (tenths of mm)
  - `TMAX`: Maximum temperature (tenths of degrees C)
  - `TMIN`: Minimum temperature (tenths of degrees C)

```r
# provide NOAA key
options(noaakey = Sys.getenv("NOAA_NCDC_CDO_token"))

# Specify datasetid and station
stn          <- "GHCND:USW00012842" # TAMPA INTERNATIONAL AIRPORT, FL US
stn_csv      <- here("data/tpa_ghcnd.csv")
stn_meta_csv <- here("data/tpa_meta.csv")

if (!file.exists(stn_meta_csv)){
  # cache station metadata since timeout from Github Actions
  stn_meta <- ncdc_stations(
    datasetid = "GHCND",
    stationid = stn)
  write_csv(stn_meta$data, stn_meta_csv)
}
read_csv(stn_meta_csv)
```

```
# A tibble: 1 x 9
  elevation mindate    maxdate    latitude name               datacoverage id
      <dbl> <date>     <date>        <dbl> <chr>                     <dbl> <chr>
1       1.8 1939-02-01 2024-01-09     28.0 TAMPA INTERNATION~            1 GHCN~
# i 2 more variables: elevationUnit <chr>, longitude <dbl>
```

```r
if (!file.exists(stn_csv)){

  date_beg <- stn_meta$data$mindate
  date_end <- stn_meta$data$maxdate
  max_rows <- 1000
  vars     <- c("PRCP","TMIN","TMAX")

  n_vars     <- length(vars)
  days_batch <- floor(max_rows / n_vars)
  dates <- unique(c(
    seq(
      ymd(date_beg),
      ymd(date_end),
      by = glue("{days_batch} days")),
    ymd(date_end)))

  n_i <- length(dates) - 1
  for (i in 1:n_i){
```

```r
  # for (i in 14:n_i){
  date_beg <- dates[i]
  if (i == n_i){
    date_end <- dates[i+1]
  } else {
    date_end <- dates[i+1] - days(1)
  }
  print(glue("{i} of {n_i}: {date_beg} to {date_end}  ~  {Sys.time()}"))

  # retry if get Error: Service Unavailable (HTTP 503)
  o           <- NULL
  attempt     <- 1
  attempt_max <- 10
  while (is.null(o) && attempt <= attempt_max) {
    if (attempt > 1)
      print(glue("  attempt {attempt}", .trim = F))
    attempt <- attempt + 1
    try(
      o <- ncdc(
        datasetid  = "GHCND",
        stationid  = stn,
        datatypeid = vars,
        startdate  = date_beg,
        enddate    = date_end,
        limit      = max_rows) )
  }

  if (i == 1) {
    df <- o$data
  } else {
    df <- rbind(df, o$data)
  }
}
stopifnot(duplicated(df[,1:2])|> sum() == 0)

df <- df |>
  mutate(
    date     = as.Date(strptime(
      date, "%Y-%m-%dT00:00:00")),
    datatype = recode(
      datatype,
      PRCP = "precip_mm",
```

```
        TMIN = "temp_c_min",
        TMAX = "temp_c_max"),
      value    = value / 10) |>
    select(
      -station, # station         : all "GHCND:USW00012842"
      -fl_m,    # measurement flag: 3,524 are "T" for trace
      -fl_t,    # time        flag: all "2400"
      -fl_q)    # quality     flag: all ""

  write_csv(df, stn_csv)
}

d <- read_csv(stn_csv)

d |>
  select(date, datatype, value) |>
  filter(datatype %in% c("temp_c_min","temp_c_max")) |>
  pivot_wider(
    names_from  = datatype,
    values_from = value) |>
  dygraph(main = "Daily Temperature (ºC)") |>
  dyOptions(
    colors = brewer.pal(5, "YlOrRd")[c(5,3)]) |>
  dySeries("temp_c_min", label = "min") |>
  dySeries("temp_c_max", label = "max")
```

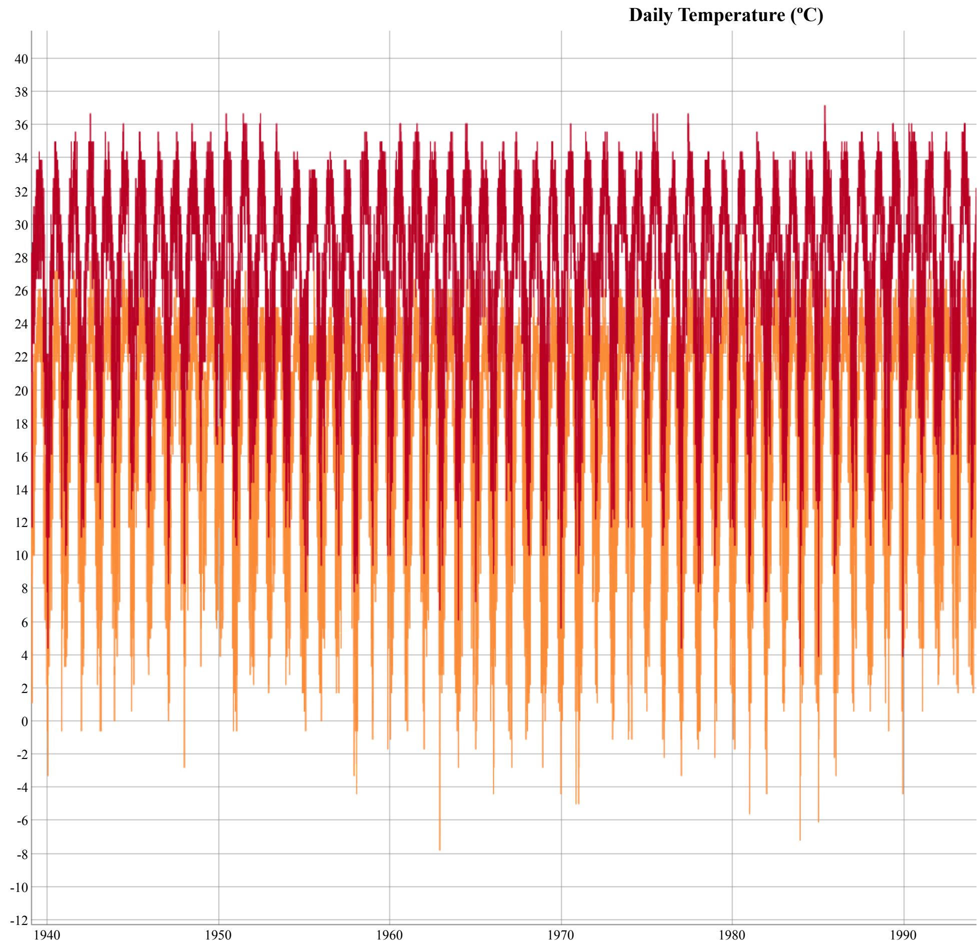

Daily Temperature (ºC)
40
38
36
34
32
30
28
26
24
22
20
18
16
14
12
10
8
6
4
2
0
-2
-4
-6
-8
-10
-12
1940
1950
1960
1970
1980
1990

Figure 2.1

```r
d |>
  select(date, datatype, value) |>
  filter(datatype %in% c("precip_mm")) |>
  pivot_wider(
    names_from  = datatype,
    values_from = value) |>
  dygraph(main = "Daily Precipitation (mm)") |>
  dySeries("precip_mm", label = "precip")
```

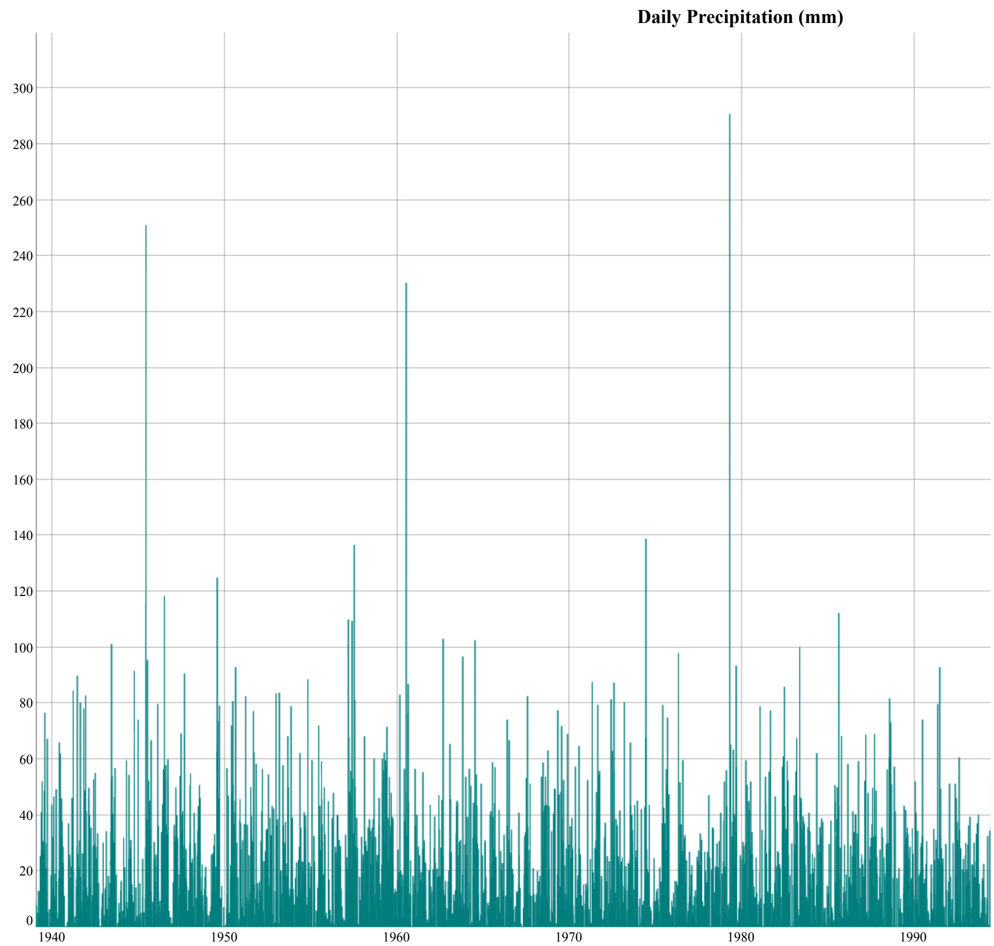

Daily Precipitation (mm)
300
280
260
240
220
200
180
160
140
120
100
80
60
40
20
0
1940
1950
1960
1970
1980
1990

Figure 2.2

TODO: - trend analysis. e.g. NOAA's Climate at a Glance. Typically based on the last 30 years, but here we've got back to 1939-02-01 so almost 100 years. Keep it 5 years and see how rate changing over time.

- severe weather events? "Sea-level rise exponentially increases coastal flood frequency Mohsen taherkhani"

2.2.1.2 Satellite

2.3 Precipitation

Materials:

- "RAIN AS A DRIVER" in tbep-os-presentations/state_of_the_bay_2023.qmd
- Precipitation - NEXRAD QPE CDR | National Centers for Environmental Information (NCEI)

```
librarian::shelf(
  dplyr, here, leaflet,
  # mapview,
  readxl, sf, tbep-tech/tbeptools)
# register with renv
library(dplyr)
library(here)
library(leaflet)
# library(mapview)
library(readxl)
library(sf)
library(tbeptools)

# from SWFWMD grid cells, use only if interested in areas finer than TB watershed
# this currently gets the same data as the compiled spreadsheet
grd <- st_read(here('../tbep-os-presentations/data/swfwmd-GARR-gisfiles-utm/swfwmd_pixel_2_u
# mapView(grd)

tbgrdcent <- grd %>%
  st_transform(crs = st_crs(tbshed)) %>%
  st_centroid() %>%
  .[tbshed, ]

# unzip folders
```

```r
loc <- here('../tbep-os-presentations/data/swfwmd_rain')
# files <- list.files(loc, pattern = '.zip', full.names = T)
# lapply(files, unzip, exdir = loc)

# read text files
raindat <- list.files(loc, pattern = '19.*\\.txt$|20.*\\.txt$', full.names = T) %>%
  lapply(read.table, sep = ',', header = F) %>%
  do.call('rbind', .) %>%
  rename(
    'PIXEL' = V1,
    'yr' = V2,
    'inches' = V3) %>%
  filter(PIXEL %in% tbgrdcent$PIXEL)

# ave rain dat
raindatave <- raindat %>%
  summarise(
    inches = mean(inches, na.rm = T),
    .by = 'yr')

##
# use compiled SWFWMD data

# # https://www.swfwmd.state.fl.us/resources/data-maps/rainfall-summary-data-region
# # file is from the link "USGS watershed"
# download.file(
#   'https://www4.swfwmd.state.fl.us/RDDataImages/surf.xlsx?_ga=2.186665249.868698214.170592
#   here('data/swfwmdrainfall.xlsx'),
#   mode = 'wb'
#   )

raindatave_url <- "https://www4.swfwmd.state.fl.us/RDDataImages/surf.xlsx"
dir.create(here('data/swfwmd.state.fl.us'))
raindatave_xl <- here('data/swfwmd.state.fl.us/surf.xlsx')

download.file(raindatave_url, raindatave_xl)
read_excel(raindatave_xl)

download.file(raindatave_url, here('data/swfwmdrainfall.xlsx'), mode = 'wb')

raindatave <- read_excel(
  raindatave_xl, sheet = 'ann-usgsbsn', skip = 1) %>%
```

```
  filter(Year %in% 1975:2023) %>%
  select(
    yr = Year,
    inches = `Tampa Bay/Coastal Areas`
  ) %>%
  mutate_all(as.numeric)

raindatave_now <-
readxl::read_excel()

raindatave <- read_excel(here('data/swfwmdrainfall.xlsx'), sheet = 'ann-usgsbsn', skip = 1) %
  filter(Year %in% 1975:2023) %>%
  select(
    yr = Year,
    inches = `Tampa Bay/Coastal Areas`
  ) %>%
  mutate_all(as.numeric)

# ave chldat
chlave <- anlz_avedat(epcdata) %>%
  .$ann %>%
  filter(var == 'mean_chla') %>%
  summarise(
    chla = mean(val, na.rm = T),
    .by = 'yr'
  ) %>%
  filter(yr >= 1975)

toplo <- inner_join(chlave, raindatave, by = 'yr')

p1 <- ggplot(raindatave, aes(x = yr, y = inches)) +
  geom_line() +
  geom_point() +
  geom_point(data = raindatave[chlave$yr == 2023, ], col = 'red', size = 2) +
  theme_minimal() +
  theme(
    panel.grid.minor = element_blank(),
  ) +
  labs(
    x = NULL,
    y = 'Annual rainfall (inches)',
    title = 'Annual rainfall',
```

```
    subtitle = 'Tampa Bay watershed, 1975 - 2023'
  )

p2 <- ggplot(chlave, aes(x = yr, y = chla)) +
  geom_line() +
  geom_point() +
  geom_point(data = chlave[chlave$yr == 2023, ], col = 'red', size = 2) +
  theme_minimal() +
  theme(
    panel.grid.minor = element_blank(),
  ) +
  labs(
    x = NULL,
    y = 'Chlorophyll-a (ug/L)',
    title = 'Annual mean chlorophyll-a',
    subtitle = 'All segments, 1975 - 2023'
  )

p3 <- ggplot(toplo, aes(x = inches, y = chla)) +
  geom_text_repel(aes(label = yr), point.size = NA, segment.size = NA) +
  geom_label_repel(data = toplo[toplo$yr == 2023, ], aes(label = yr), color = 'red', point.si
  geom_smooth(formula = y ~ x, method = 'lm', se = F, color = 'red') +
  # geom_segment(aes(x = 45, xend = 40, y = 4.86, yend = 4.86), color = 'red', arrow = arrow
  theme_minimal() +
  theme(
    panel.grid.minor = element_blank(),
  ) +
  labs(
    x = 'Annual rainfall (inches)',
    y = 'Chlorophyll-a (ug/L)',
    title = 'Annual mean chlorophyll-a vs. rainfall',
    caption = 'Data from EPCHC, SWFWMD'
  )

p <- (p1 / p2) | p3
p
```

2.3.1 rNOMADS

```
# librarian::shelf(rNOMADS)
```

2.3.2 prism

- HMS: Hydrologic Micro Services | United States Environmental Protection Agency | US EPA

  The Parameter-elevation Relationship on Independent Slopes Model (PRISM) is a combined dataset consisting of ground gauge station and RADAR products. The data is on a 4km grid resolution covering the contiguous United States. Data is available from 1981 to present.PRISM data are reported in GMT (UTC). PRISM provides daily average temperature and dew-point temperature data. Relative humidity is calculated using a version of the August-Roche-Magnus equation as follows): `RH = 100*(EXP((17.625*TD)/(243.04+TD))/EXP((17.625*T)/(243.04+T)))` where, `RH` is % relative humidity, `TD` is dew-point temperature (celsius), and `T` is air temperature (celsius).

- AHPS Precipitation Analysis

  > "Normal" precipitation is derived from PRISM climate data, created at Oregon State University. The PRISM gridded climate maps are considered the most detailed, highest-quality spatial climate datasets currently available.

- prism R package

Parameter	Description
`tmin`	Minimum temperature
`tmax`	Maximum temperature
`tmean`	Mean temperature (`tmean == mean(tmin, tmax)`)
`tdmean`	Mean dew point temperature
`ppt`	Total precipitation (rain and snow)
`vpdmin`	Daily minimum vapor pressure deficit
`vpdmax`	Daily maximum vapor pressure deficit

  Data are at 4km resolution, except for the normals which can also be downloaded at 800m resolution.

Temporal data availability:

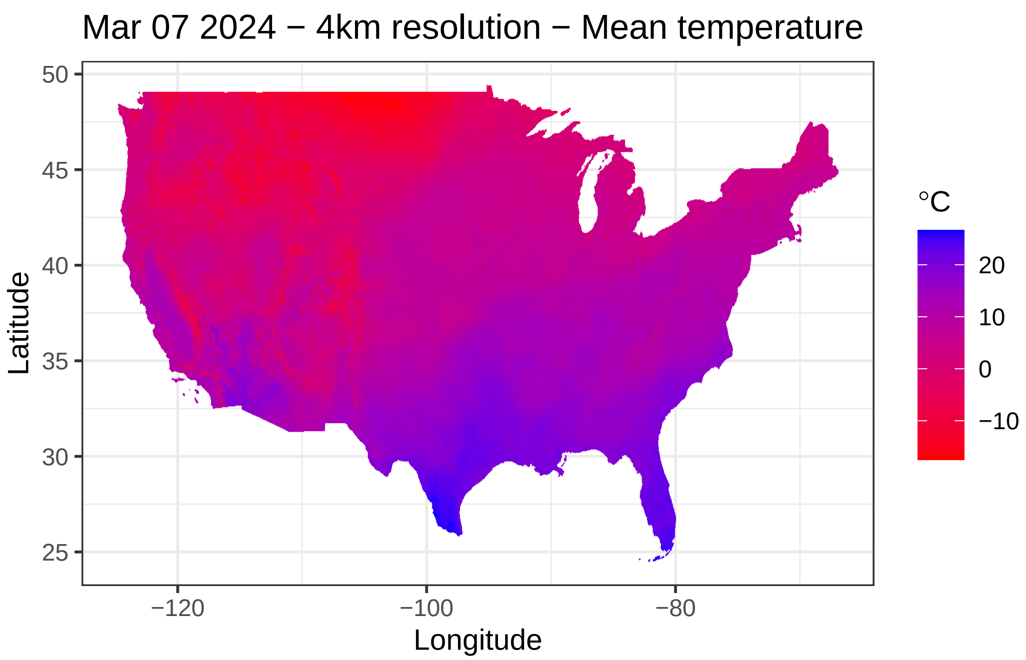
- **Recent**
  1981 to present
  daily, monthly, annual data
- **Historical**
  1895 through 1980
  complete monthly and annual data by year
- **Normals**
  30-year normals daily, monthly, and annual normals, each as a single grid The 30 year PRISM normal from 1981-2010 is used for precipitation analysis since 2004. Prior to 2004 the 30 year PRISM normal from 1961-1990 is used.

```r
# librarian::shelf(glue, here, leaflet, lubridate,
# # mapview, stars,
# prism, sf, stringr, tbep-tech/tbeptools, terra)

# renv register libraries
library(glue)
library(here)
library(leaflet)
library(lubridate)
# library(mapview)
library(prism)
# library(stars)
library(sf)
library(stringr)
library(tbeptools)
library(terra)

dir_prism <- here("tmp/prism")
dir.create(dir_prism, showWarnings = F)
prism_set_dl_dir(dir_prism)

# get_prism_dailys(
#   type    = "tmean",
#   minDate = today() - days(8),
#   maxDate = today() - days(1),  # up to yesterday
#   keepZip = F)
# prism_archive_clean("tmean")
# https://prism.nacse.org/downloads/
#   https://prism.nacse.org/documents/PRISM_downloads_web_service.pdf
#     http://services.nacse.org/prism/data/public/4km/tmin/20090405
```

```
# get_prism_dailys(
#   type    = "tmin",
#   minDate = "2024-01-01",
#   maxDate = "2024-03-07",
#   keepZip = F)
#
# get_prism_dailys(
#   type    = "tmax",
#   minDate = "2024-01-01",
#   maxDate = "2024-03-07",
#   keepZip = F)

# Get monthly (every month) and annual 30-year normals for precipitation
# get_prism_normals(
#   type       = "ppt",
#   resolution = "800m",
#   mon        = 1:12,
#   annual     = TRUE,
#   keepZip    = FALSE)

# Plot the January 30-year average temperatures
# grab only the first value, just in case multiple values are returned
# pd_tmean_day <- prism_archive_subset(
#   "tmean", "daily", dates = "2024-03-07")
# Error in pd_image(pd_tmean_day, col = "redblue") :
#   You can only quick image one file at a time.
# > pd_tmean_day
# [1] "PRISM_tmean_early_4kmD2_20240307_bil"       "PRISM_tmean_provisional_4kmD2_20240307_bi
# prism_archive_clean("tmean", "daily")  # 'cleans' the prism download data by removing early
period <- "daily"
var    <- "tmean"
date   <- "2024-03-07"
get_prism_dailys(type=var, dates=date, keepZip=F)
```

```
  |
  |                                                                      |   0%
  |
  |======================================================================| 100%
```

```r
pd_tmean_day <- prism_archive_subset(var, period, dates = date)
pd_image(pd_tmean_day, col="redblue")
```



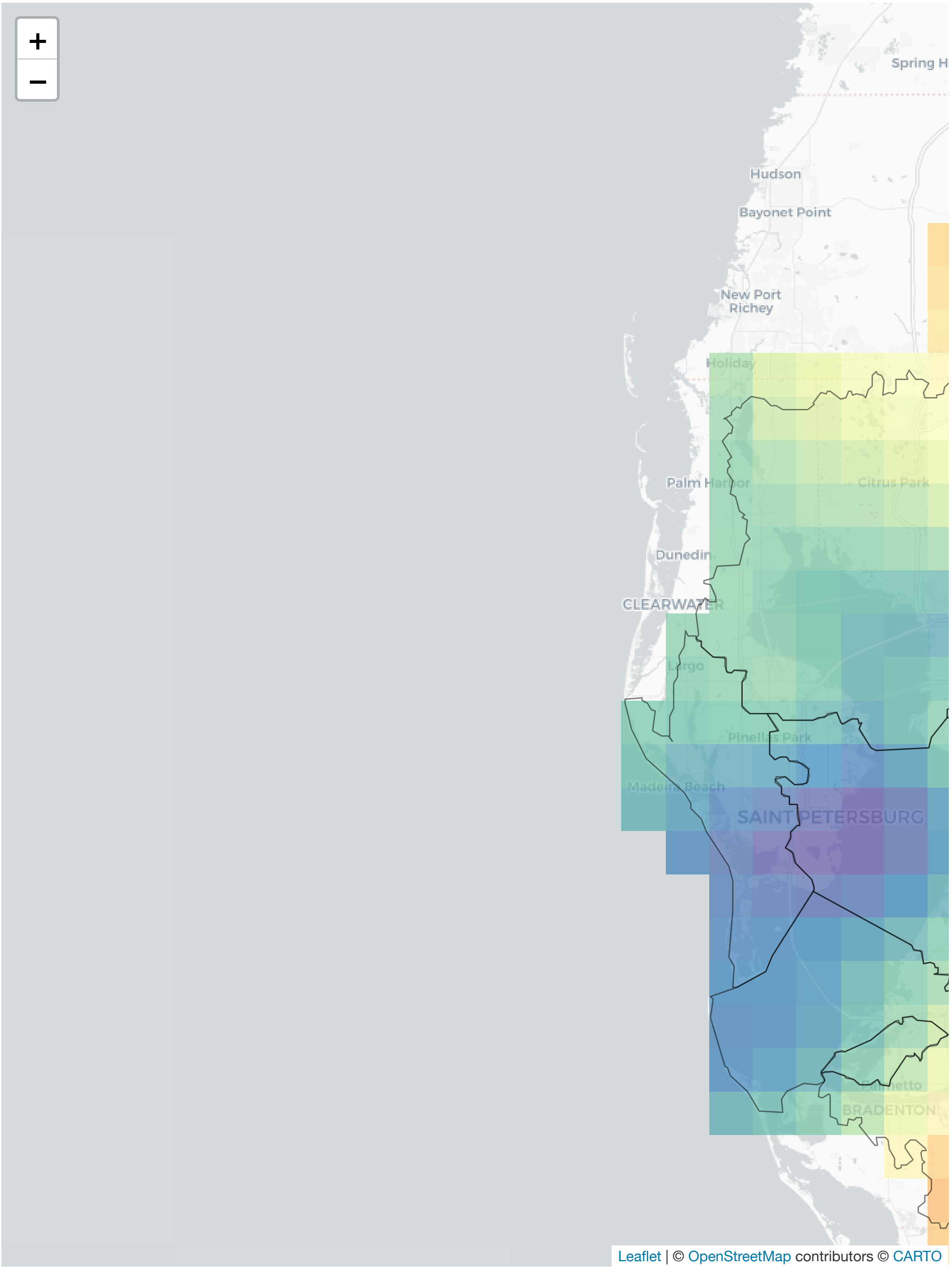

```r
# pd_ppt_nrml <- prism_archive_subset(
#   "ppt", "monthly normals",
#   mon = 1, resolution = "800m")
# (path_ppt_nrml <- pd_to_file(pd_ppt_nrml))

# raster stack ----
var    <- "tmean"
period <- "daily"

get_prism_dailys(
  type     = var,
  minDate = today() - days(3),  # last 3 days
  maxDate = today() - days(1),  # up to yesterday
  keepZip = F)
```

```
  |
  |                                                                      |   0%
  |
```

```
  |=======================                                               |  33%
  |
  |===============================================                       |  67%
  |
  |======================================================================| 100%
```

```
prism_archive_clean(var, period)

pd       <- prism_archive_subset(var, period)
pd_dates <- pd_get_date(pd)

r <- pd_stack(pd) |>
  rast()
# tbshed_pd <- tbshed |>
tbshed_pd <- tbsegshed |>
  st_transform(crs(r, proj=T))
# sf broken, so Warning: Cannot find proj.db
r_tb <- r |>
  crop(tbshed_pd, mask = T, touches = T) |>
  trim()

# TODO: summarize by tbeptools::tbsegshed, zip code

# r_tif <- here("tmp/prism", glue("tmean_20240307_bil")
# terra::writeRaster(r_tb, , overwrite = T)

date <- max(pd_dates)
i    <- which(pd_dates == date)
lyr  <- glue("{var}<br>{period}<br>{date}")

# terra::writeRaster(r_tb[[i]], here("tmp", glue("{var}_{date}.tif")), overwrite = T)

# mapView(r_tb[[i]], layer.name = lyr) +
#   mapView(tbshed_pd)
# sf broken, so Error: Cannot find proj.db

library(leaflet)
leaflet() |>
  addProviderTiles(
    providers$CartoDB.Positron) |>
  addRasterImage(
    r_tb[[i]], opacity = 0.7) |>
```

```
  addPolygons(
    data = tbsegshed,
    fillOpacity = 0,
    color = "black", weight = 1)
```


+
−
Spring H
Hudson
Bayonet Point
New Port Richey
Holiday
Palm Harbor
Citrus Park
Dunedin
CLEARWATER
Largo
Pinellas Park
Madeira Beach
SAINT PETERSBURG
BRADENTON
Leaflet | © OpenStreetMap contributors © CARTO

TODO:

- ☐ compare these precip data w/ Water District data to make case for using PRISM data

Questions:

1. Given variability within each polygon, which of these products shall we use to plot: min(min_temp), mean(mean_temp), max(max_temp); min(mean_temp), mean(mean_temp), max(max_temp)?

2.3.3 Communicating results

- Choi et al. (2024) North-South disparity in impact of climate change on "outdoor days". *Journal of Climate*
  - news summary: A new way to quantify climate change impacts: "Outdoor days" | MIT News | Massachusetts Institute of Technology
  - Shiny app: California Outdoor Days | Eltahir Research Group

2.4 Sea Level Rise

Sea level rise occurs from principally two sources: 1) thermal expansion; and 2) freshwater inputs from glacial melting. Data for these trends can be obtained from NOAA's Sea Level Trends (Figure 2.3)

Types of data:

1. Observed (past, present) - tide gauge - satellite, e.g. Laboratory for Satellite Altimetry / Sea Level Rise
   - Level-3 products distributed through NOAA CoastWatch (Sea Level Anomaly and along-track altimetry)
2. Projected (future). modeled

2.4.1 Gauges

- PORTS: Tampa Bay PORTS - NOAA Tides & Currents

```
librarian::shelf(
  ropensci/rnoaa)
# register with renv
library(rnoaa)
```

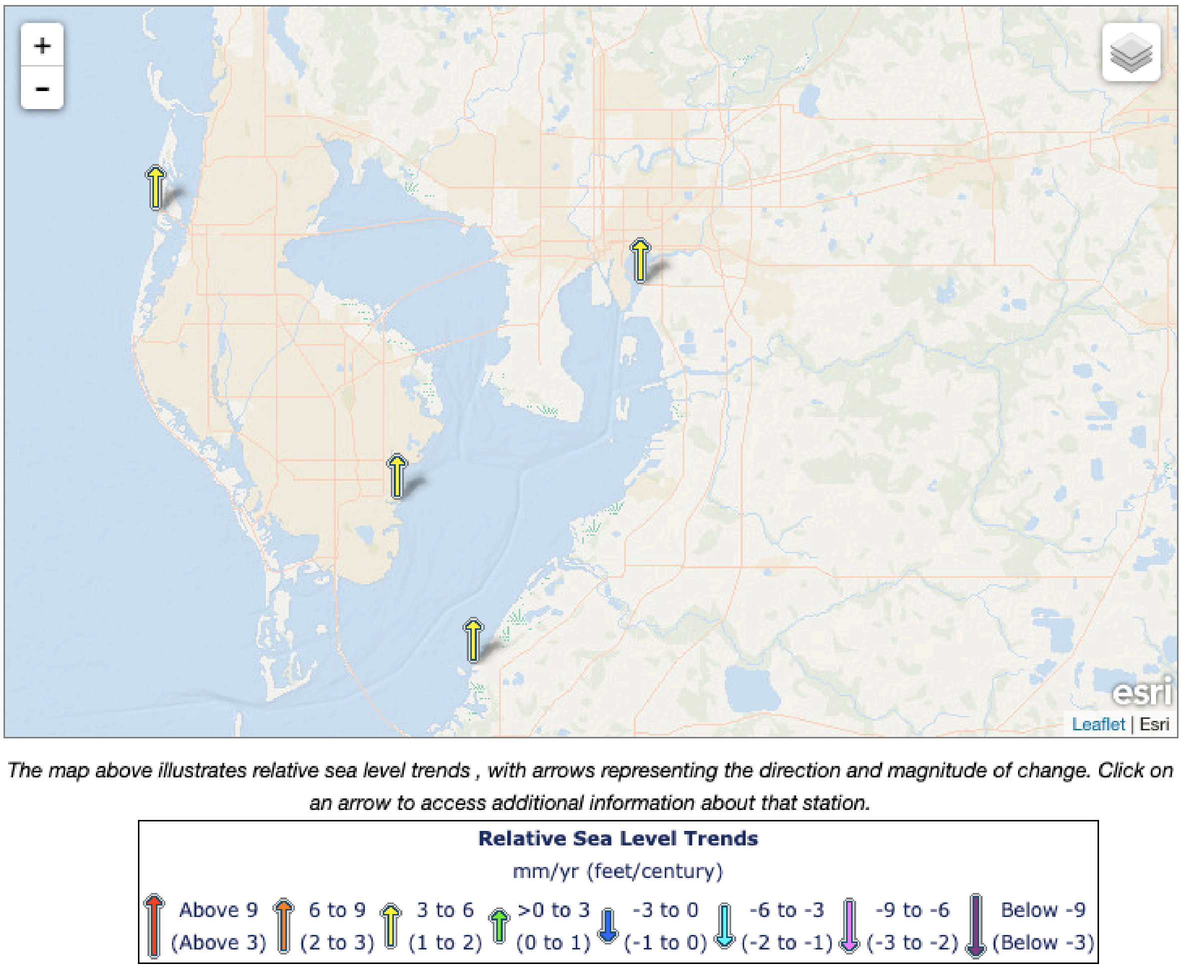


Figure 2.3: Screenshot of NOAA's Sea Level Trends zoomed into the Tampa Bay.

2.4.2 Satellite

- NOAA / NESDIS / STAR - Laboratory for Satellite Altimetry / Sea Level Rise

```
slr_nc    <- here("data/slr/slr_map_txj1j2.nc")
r_slr_gcs <- rast(slr_nc)  # 0.5 degree resolution
r_slr_mer <- projectRasterForLeaflet(r_slr_gcs, method="bilinear")

b <- st_bbox(tbsegshed)
r_slr_tb_mer <- rast(slr_nc) |>
  crop(b) # |>
  # projectRasterForLeaflet(method="bilinear")
# only one value for Tampa Bay extracted at 0.5 degree resolution
# values(r_slr_tb_mer, mat=F, na.rm=T)  # 5.368306

b <- st_bbox(tbshed)
plet(r_slr_mer, tiles=providers$Esri.OceanBasemap) |>
  addProviderTiles(providers$CartoDB.DarkMatterOnlyLabels) |>
  addPolygons(data = tbsegshed) |>
  fitBounds(
    lng1 = b[["xmin"]], lat1 = b[["ymin"]],
    lng2 = b[["xmax"]], lat2 = b[["ymax"]])
```


+
−
Spring H
Hudson
Bayonet Point
New Port Richey
Holiday
Palm Harbor
Citrus Park
Dunedin
CLEARWATER
Largo
Pinellas Park
Madeira Beach
SAINT PETERSBURG
Sea_level_trends
15
10
5
0
-5
Leaflet | Tiles © Esri — Sources: GEBCO, NOAA, CHS, OSU, UNH, CSUMB, National Geographic, DeLorme, NAVTEQ, and Esri, © OpenStreetMap contributors © CARTO

Figure 2.4

2.5 Severe Weather

- SWDI vignette
- rnoaa

References

Beck, Marcus W., G. E. Raulerson, M. C. Burke, J. Whalen, S. Scolaro, and E. T. Sherwood. 2021. "Tampa Bay Estuary Program: Data Management Standard Operating Procedures." St. Petersburg, Florida.

Beck, Marcus W., Meagan N. Schrandt, Michael R. Wessel, Edward T. Sherwood, Gary E. Raulerson, Adhokshaja Achar Budihal Prasad, and Benjamin D. Best. 2021. "Tbeptools: An R Package for Synthesizing Estuarine Data for Environmental Research." *Journal of Open Source Software* 6 (65): 3485. https://doi.org/10.21105/joss.03485.

Burke, M., and M. Amaral. 2020. "2021-2025 Strategic Plan." St. Petersburg, Florida.