# Automated and operational access to environmental data for Alaska’s management areas

Jordan T Watson and Matthew W Callahan

# Abstract

The proliferation of operational satellite data has facilitated downstream data products catered towards specific fisheries applications in near real-time. The Alaska Fisheries Science Center (AFSC) and Alaska Fisheries Information Network (AKFIN) have utilized such data accessibility to develop processes that streamline use of satellite sea surface temperature (SST) data. Here we briefly describe data products used and present two such processes: 1) aggregated SST by management regions of interest and 2) a suite of fishery-dependent data linked with spatially-explicit SST. To aggregate SST data, we clipped gridded SST data to regions of interest for Alaska fisheries management. Full gridded data sets apportioned to management can be queried from the AKFIN database. Alternatively, aggregated data products (e.g., time series of SST for individual NMFS regions or ecosystem areas) can be accessed via a custom AKFIN SST web service. We demonstrate several queries of the web service and illustrate how this product can yield seamless integration with downstream analyses. For fisheries dependent data (i.e. fish-ticket or observer), sea surface temperature (SST) data were linked to all fish tickets and observer in the Oracle backend from 2002 - present (more than one million records), and new data are automatically matched each day. We provide examples of how these processes facilitate efficient use of SST data, and describe opportunities for expansion of these services.

# Introduction

The role of oceanographic and other environmental or ecosystem parameters on the productivity of the world’s fish stocks has long been established. Increasingly, such parameters are explicitly incorporated into fisheries stock assessments ([Holsman et al. 2016](#ref-Holsman2016), [**Marshall2019?**](#ref-Marshall2019)), risk assessments ([Gaichas et al. 2014](#ref-Gaichas2014)); ecosystem reports ([**Ferriss2020?**](#ref-Ferriss2020), [**Ortiz2020?**](#ref-Ortiz2020), [**Siddon2020?**](#ref-Siddon2020)), or other documents used by the U.S. Regional Fishery Management Councils to guide decision making. Meanwhile, a growing trend in the development of dynamic ocean management tools seeks to incorporate environmental information in near real-time to inform stakeholders for bycatch avoidance ([Hazen et al. 2018](#ref-Hazen2018), [**Breece2021?**](#ref-Breece2021)), harmful algal blooms ([Harley et al. 2020](#ref-Harley2020)), avoiding interactions with protected species (<https://oceanview.pfeg.noaa.gov/whale_indices/>), and more. Thus, as NOAA moves towards a broader adoption of ecosystem-based fisheries management and dynamic ocean management, the accessibility of ecosystem information becomes increasingly critical.

One of the most fundamental ecosystem parameters considered in fisheries is water temperature. Temperature regulates the timing and intensity of primary production, which has ripple effects on secondary producers and on to higher trophic levels. Temperature directly impacts fish growth and other metabolic processes in addition to regulating the location and abundance of prey. Thus, for most mobile fish species, temperature often defines the habitat of the species, and subsequently, the location of the fishing fleets that target them ([Rogers et al. 2019](#ref-Rogers2019), [**Haynie2012?**](#ref-Haynie2012), [**Watson2018?**](#ref-Watson2018)).

As global climate changes, water temperatures have been among the most easily measured metrics by which to understand how ocean ecosystems are responding. Broad warming trends are leading to poleward shifts in the distributions of fish species and the fleets that target them ([Kotwicki and Lauth 2013](#ref-Kotwicki2013), [Rogers et al. 2019](#ref-Rogers2019), [Pinsky et al. 2020](#ref-Pinsky2020), [Fredston et al. 2021](#ref-Fredston2021)), while anomalously warm periods or marine heatwaves are driving protracted impacts on ecosystems ([**Suryan2021?**](#ref-Suryan2021)) and commercial fish stocks ([Barbeaux et al. 2020](#ref-Barbeaux2020)). Such dynamics underscore the need for reliable access to near real-time water temperature data.

Satellite-derived sea surface temperature data are well validated have been available since the early 1980s (Minnett et al., 2019). A proliferation of new technologies, sensors, and data products have led to increasingly frequent and spatially resolved information with latencies as little as one day (Liu et al., 2015; Maturi et al., 2017; Minnett et al., 2019). Moreover, the development of programs like NOAA’s CoastWatch and data technologies like Environmental Research Division’s data access program (ERDDAP) servers (Simons 2020) have facilitated easier access to these data worldwide in near real-time and via a suite of data formats. While such technologies have improved data access, challenges still exist for some end users due to the large file sizes of high spatial and temporal resolution data sets, difficulty subsetting data within irregular polygons (custom spatial strata), and the need for data infrastructure that supports operationalization and automation of data ingestion (Welch et al., 2019).

After assessing the needs of a suite of fisheries biology, stock assessment, and socio-ecological modeling efforts at the AFSC, we developed an automated and operational framework for serving satellite environmental data products for a suite of spatial strata used for fisheries management and research in Alaska. These are products initiated by AFSC requests, and developed by AKFIN and AFSC staff in the AKFIN database environment. We describe the data used, the process for joining the data to spatial strata, data access through customized web services (data queries via URL), and backend database merges with fishery dependent data (e.g. observer and fish ticket data).

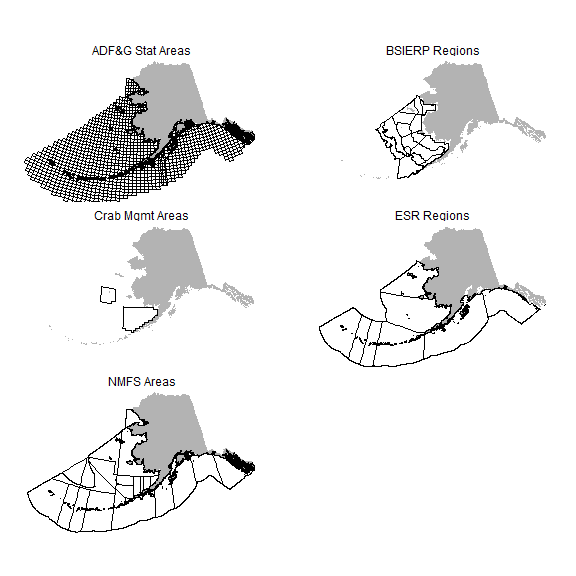
# Methods & Results

*Satellite data*  
For this study, three daily satellite sea surface temperature (SST) products were used, all of which were accessed via NOAA ERDDAP servers ([Simons 2020](#ref-Simons2020)) and downloaded as netCDF files within the Oracle database backend at the Alaska Fisheries Information Network (AKFIN), maintained by the Pacific States Marine Fisheries Commission. The SST data are publicly available but by ingesting them into the AKFIN backend, they can be seamlessly merged, behind the NOAA firewall, with confidential fishery-dependent data sets like observer data, vessel monitoring system (VMS) data, and fish tickets. These SST products provide gap-free data each day with a 1-2 day latency period. The data sets used are the NOAA Coral Reef Watch (CRW), NASA Jet Propulsion Laboratory Multi-Scale Ultra-High Resoultion (MUR), and NOAA National Center for Environmental Information Optimal Interpolation (OI) SSTs (Table 1). The National Environmental Satellite Data and Information Service (NESDIS) recommends the CRW SST product and this is the primary dataset used in this study. The data sets vary in spatial resolution and time-span. MUR SST has the finest spatial resolution while OI SST is the most course, and OI SST extends furthest back while MUR has the shortest duration available. The CRW data is intermediate between data are obtained from the NOAA Coral Reef Watch Program intermediate between MUR and OI SST with a 0.05° x 0.05° (5 km) spatial grid from 1985-present.

All three data sets have native formats with longitudes ranging from -180 to +180. Because the spatial extent for Alaska waters includes the International Date Line, the daily data are downloaded via two separate operations each day. One operation downloads the negative longitude data from 46°N to 68.8°N and -180°E to -130°E and the second operation downloads the positive longitude data from 47.5°N to 60.0°N and 164°E to 180°E. These downloads are merged and then clipped to spatial regions of interest within the exclusive economic zone surrounding Alaska, yielding 212,813 SST records per day for the CRW data set.

*Spatial strata* State and Federal waters of Alaska include numerous spatial strata that are relevant to fisheries management, ecology, and individual species distributions. For example, the Alaska Department of Fish & Game (ADF&G) divides Alaskan waters into nearly 1,800 statistical areas, many of which are 0.5° latitude by 1.0° longitude boxes. Meanwhile, the National Marine Fisheries Service (NMFS) divides the same waters into only 25 management areas. However, these regulatory strata are inconsistent with ecological stratifications (Eastern Bering Sea, Gulf of Alaska, and the Aleutian Islands) identified for the same waters. These ecosystem regions, even when subdivided, do not necessarily align with spatial strata identified for individual fish or crab stocks, so stock assessment scientists and fishery managers are often interested in yet further customized spatial boundaries. Thus it is not surprising that different users of environmental information like SST may want those data aggregated or clipped to a different (or multiple) spatial boundaries.

To develop operational data products across Alaska’s suite of spatial strata, we undertook extensive point-in-polygon geoprocessing operations to apportion the individual latitude-longitude coordinates for both the MUR and CRW SST spatial grids to each of the polygons from a suite of shapefiles (ADF&G management areas, NMFS management areas, Ecosystem regions [from NMFS Ecosystem Status Reports], Bering Sea Integrated Ecosystem Research Program [BSIERP] regions, Bristol Bay red king crab management areas, and St. Matthew blue king crab management areas) (Fig. 1). To avoid repeating the computationally intensive point-in-polygon operations, we created spatial lookup tables that are stored in the backend of the AKFIN Oracle database system. Thus, as data are downloaded daily from ERDDAP servers across the spatial extent of Alaska’s waters, each SST record is matched via a database join on latitude and longitude to the spatial strata in which it falls.

 Figure 1. Spatial strata in Alaska for which sea surface temperature data have been clipped and aggregated within the AKFIN database backend. SST data for these strata can be queried and accessed several ways.

*Accessing the data* Data ingested into AKFIN can be accessed and used for operational workflows via several different methods (Figure 2). We demonstrate two general methods for accessing the data stored in AKFIN. The first method, customized web services (web APIs), is ideal for accessing time series of aggregated data (e.g. daily SST averaged across a spatial stratum or multiple spatial strata) and for queries less than about 100,000 records. This approach leverages a simplified data access point (url) that is outside of the AKFIN firewall and requires no user login. The second method, direct database access, requires a login to the AKFIN database backend and relies on SQL to extract either aggregated data summaries or larger gridded data sets (e.g., millions of data records). In the sections that follow, we demonstrate data queries using custom web services and by using direct SQL and R access. For each case, we illustrate the utility of operational data workflows by piping these data into R functions for calculating marine heatwaves (MHWs).

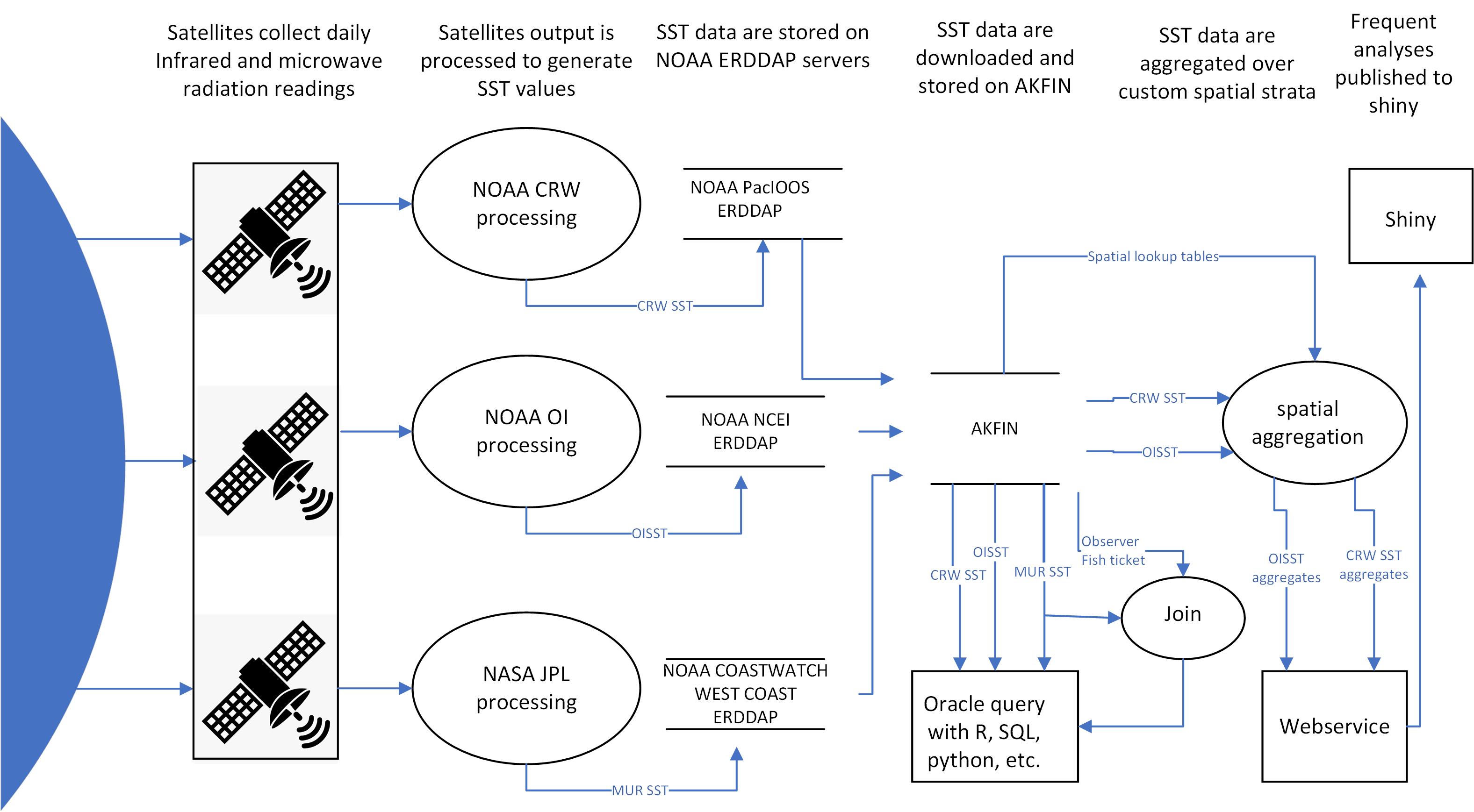


Figure 2: Data flow diagram for the ingestion, processing, and extraction of satellite sea surface temperature data within AKFIN.

*Customized Web Service (Web API)*

For queries that are likely to be repeated often or to become part of an automated process, customized web services offer a particularly efficient data access option. These web services require no accounts, no passwords, no VPN - just internet and a web service URL can be readily embedded into programming applications (e.g., R, Python). An additional convenience is that web services allow users to query time series without storing data locally which is particularly helpful for operations that would typically append data to existing files (e.g., adding a new day of data to a time series). With this tool, users can easily incorporate SST data into stock assessments and other processes.

The AKFIN web service web service enables a query of spatially aggregated CRW SST using a URL, where the URL contains the query parameters (Fig. 3). For example, in the URL “<https://apex.psmfc.org/akfin/data_marts/akmp/nmfs_area_crw_avg_sst?nmfs_area=640>,” where “nmfs\_area\_crw\_avg\_sst?” is the name of the dataset. This is the daily SST data averaged by nmfs\_area. A “?” separates the data set name from the query criteria. To query multiple areas, separate the values by a comma. Large Marine Ecosystem subregions are assessed in AFSC Ecosystem Status Reports (ESRs). The ecosystem\_sub fields available for query include the regions within the Eastern Bering Sea, Aleutian Islands, and Gulf of Alaska. Spaces in region names are filled with “%20.” For example, to query the data for the “Southeastern Bering Sea,” for example, add “ecosystem\_sub=Southeastern%20Bering%20Sea.” For the Bering Sea and Gulf of Alaska, the query filters only data where water depth is between 10 and 200m. For the Aleutian Islands, a depth filter is not implemented. Depth limits reflect preferences of Ecosystem Status Report contributors (e.g. Siddon 2020) and Analysts that are interested in data for different depth ranges, custom spatial bounds, or aggregated NMFS areas are encouraged to contact the authors.

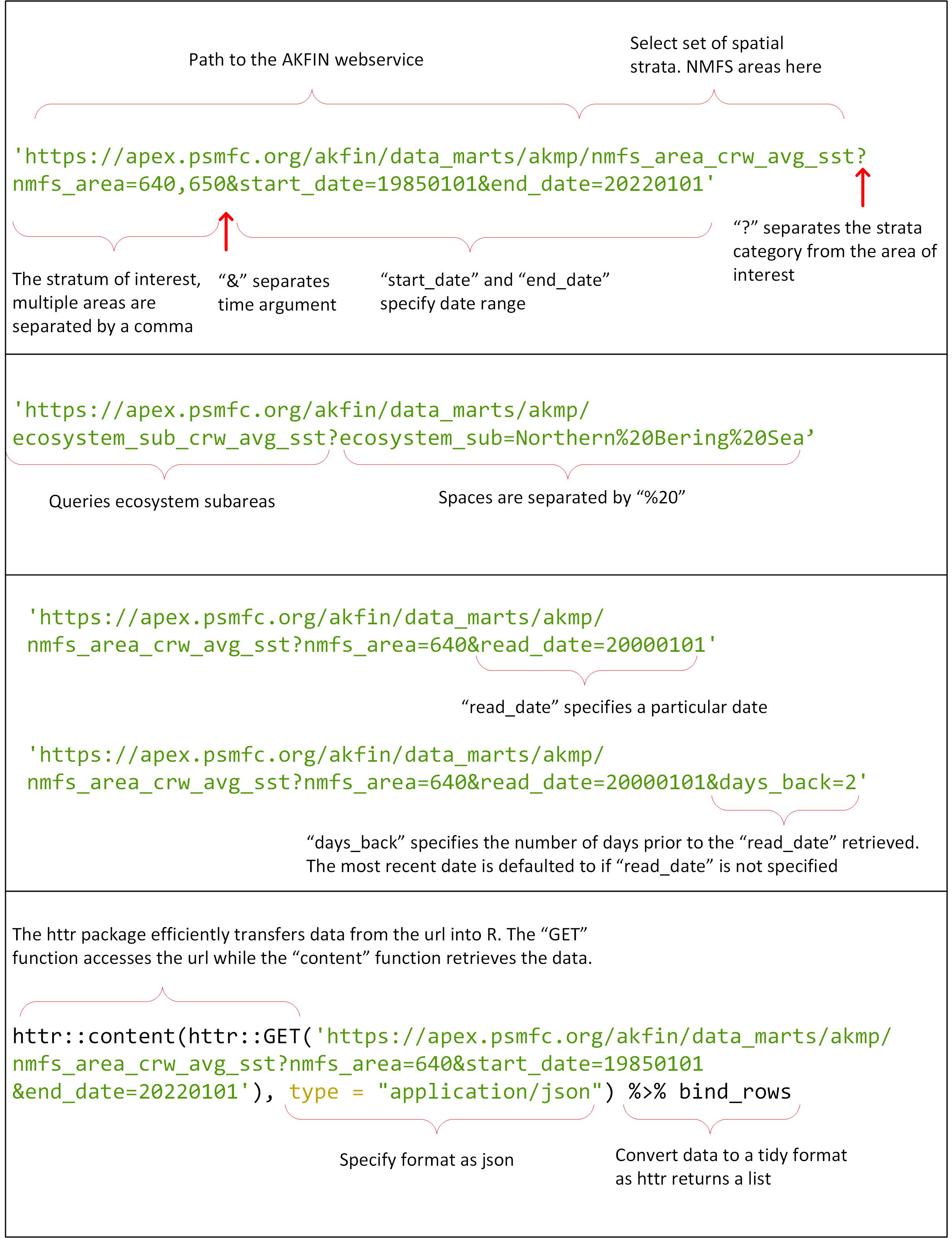


Figure 3: : Configuration of AKFIN web service URL parameters. A) Basic structure of the URL B) Multiple areas and time components. C) Additional time arguments. D) Importing webservice data into R using the httr package.

To add a time component to a query, specify “start\_date” and “end\_date,” “read\_date,” or “dates\_back” parameters. If no time argument is included, the default behavior is to pull the single most recent datum record. Time parameters should be included after spatial parameters and separated by an “&.” Most users will want the entire time series (Fig. 4), which starts on 1985-01-01. To query the entire time series, specify “start\_date” & “end\_date.” “end\_date” must be included, but if you do not know the most recent date of the time series, you can choose an end date some time in the future and it will query all of the data that exist. The full time series yields more than 13,000 rows of data per area (i.e., daily data from 1985-01-01 to present). Dates are queried in the “yyyymmdd” format, for example “19871214” queries December 14, 1987. The “read\_date” argument retrieves data from any date in the time series, however, it is necessary to query the date after the desired day. The web service date format contains a time component, which is set to 12:00:00Z for each day, SST records were created after that time stamp, causing queries to return values for the previous date. Finally, a “days\_back” parameter specification allows users to query any number of days prior a date of interest. If “read\_date” is not specified, “days\_back” returns the most recent SSTs.

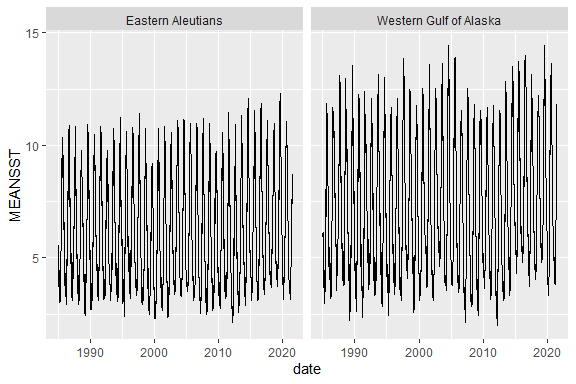


Figure 4. SST for the Western Gulf of Alaska and Eastern Aleutains areas 1985-present

To access web services using R Statistical Software, we recommend the R package httr to pull data from a URL (Fig. 3), and the data format should be specified as json. The data can be saved as an object for manipulation or piped directly into downstream functions. Additional packages tidyverse and lubridate are recommended for plotting and manipulation but the object retrieved using httr could be manipulated using base R instead. Simply pasting the URL into a web browser would also display fetched data in that browser.

**heatwave calculations from web service queries** With the daily time series for a spatial stratum, implementation of the heatwaveR package ([**W.Schlegel2018?**](#ref-W.Schlegel2018)) is straightforward. We encourage readers to explore the functionality described in the heatwaveR vignettes, from which the following examples are generated . We demonstrate a few simple examples using web services to query the SST time series (Supplement). To better illustrate the heatwave categories here, we select an example from the Northern Bering Sea from 2019-01-01 to 2019-12-31 (Fig. 5).

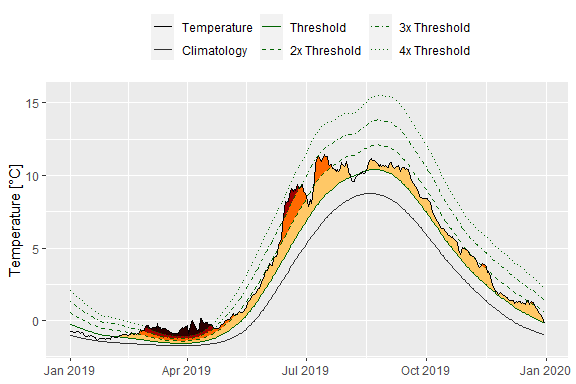


Figure 5. Marine heatwave flame plot for the Northern Bering Sea

**Oracle database queries** Oracle database queries Some data users prefer the flexibility and transparency of querying raw gridded data directly from the Oracle server. This is particularly useful for larger queries (e.g., millions of records) or for exploring data across a suite of different spatial extents (e.g., custom depth ranges, shapefiles, etc.). To query directly from the database, users will need an AKFIN database account, which can be provided by contacting the authors of this document. As we note in the web services section, such custom queries can also be automated by working with Alaska Fisheries Science Center (AFSC) and AKFIN staff.

The gridded SST data (Fig. BB) are stored within the AFSC schema on the AKFIN Oracle database and the primary key linking the lookup tables (Table 2) with the gridded data is the ID field. In the lookup table, it is simply “ID” and in the CRW SST data table it is “CRW\_ID.” Spatial columns in a query result may reveal ‘NA’ because the particular latitude - longitude coordinates shown do not fall within any spatial strata represented by those columns. The following example query demonstrates the primary key relationship between the data and lookup tables. In this case, we query SST (“TEMP”) data that fall within a crab management area and we add a field for “Year.”

SELECT read\_date, temp, TO\_CHAR(read\_date,‘YYYY’) as Year, crab FROM afsc.erddap\_crw\_sst a INNER JOIN (SELECT\* FROM afsc.erddap\_crw\_sst\_spatial\_lookup WHERE crab <> ‘NA’) b ON a.crw\_id =b.id

We provide further example code (Supplement). This section’s purpose is to orient users to the structure of the database related to the SST data and lookup tables. We assume that users interested in querying the database directly via Oracle (e.g., SQL Developer) or through odbc connections from R or Python are already acquainted with the coding and configuration settings. However, interested users can contact the authors for assistance establishing such connections or custom SQL queries.

**Matching SST data with fishery-dependent data** The above sections demonstrate access to gridded or raw SST data that are updated daily within AKFIN. In addition to raw SST data access, the daily SST data are also integrated within the AKFIN back-end to observer and fish ticket data. For both of these fishery-dependent data sets, the MUR SST data have been used, and users with access to these confidential data sets through AKFIN can find an SST field in the comprehensive\_ft and comprehensive\_haul (check table names) tables.

Observer data include latitudes and longitudes of gear deployment and retrieval locations, which are matched with the nearest gridded SST data for a given date. The temperature data are then averaged across the retrieval and deploy points to yield a single SST datum for each observed fishing event.

Fish ticket spatial data are recorded at the scale of ADF&G statistical areas (typically 0.5 degree latitude x 1.0 degree longitude), so gridded SST data cannot be matched as directly. Instead, daily SST data for all gridded locations within each statistical area (N=1758) are averaged, to yield a single daily datum for each of the statistical areas. These daily average data are then matched with the reported statistical areas on fish tickets based on the date that fishing was reported to have begun within a particular statistical area.

# Discussion

The ability to integrate environmental and fishery data sets in near real-time is fundamental to an increasing number of fishery management priorities. However, creating automated database infrastructure is beyond the expertise of most users of such data. Working with AKFIN programmers, we developed a back-end database infrastructure that automatically clips SST data to areas of interest identified by a suite of end users at the AFSC. These data can then be accessed either in gridded form, using direct database queries, or in aggregate form, using customized web services, or APIs.

The options we present each have advantages and disadvantages. The web services allow users simple and seamless access to data through a URL, which requires no login or password. **Web services have previously been used to improve data flows…** As we have demonstrated, these web services can be easily incorporated into workflows to support operational data applications, like R Shiny Apps (e.g. <https://mattcallahan.shinyapps.io/NBS_SEBS_SST_MHW/>). However, each web service URL is based on a backend SQL query that must be pre-meditated and coded by programmers. So, while the end-users do not need to code any database queries, a programmer does. Meanwhile, direct database access requires a VPN connection and a login to the AKFIN database, but once users have established this connection, they can customize any SQL queries they want using either direct Oracle access or ODBC connections through R, Python, or other data access points. This puts total control into the hands of the end-user. The goal with these combined approaches is to serve a suite of users and applications across a range of complexities of data tasks.

This document is meant to serve two primary purposes. The first is to demonstrate the functionality and access to existing environmental data products within AKFIN. The second is to give end users a sense of the types of data products and access approaches that can be requested and implemented within AKFIN. The spatial extents, satellite data sets, and web service queries demonstrated here were chosen based on previous requests or needs from individual data users at the AFSC. The framework we present uses daily sea surface temperature data but could be extended to other environmental data products like chlorophyll, wind, ROMS model extractions, or other data identified by stakeholders. The authors of this study are keen to work with end-users and AKFIN staff to connect additional data needs with AFSC end-users. Thus, we encourage data users to contact us to discuss data access, automation, and operationalization needs and interests.

#Supplement The following section provides examples of R and SQL code that utilizes the AKFIN SST webservice or AKFIN backend SST data.

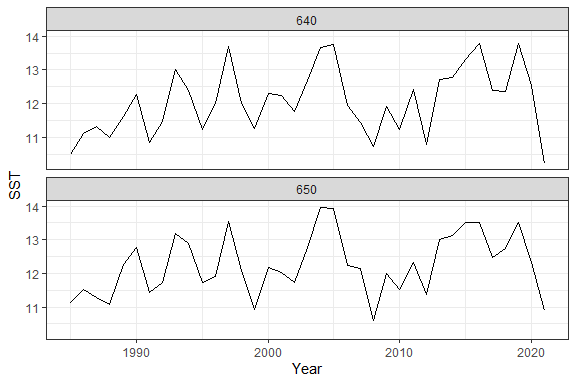
**R Code Input - 1**

#Install packages  
library(httr) # For pulling data via a URL  
library(tidyverse) # Data manipulation  
library(lubridate) # Date formatting  
library(odbc) # For connecting to oracle database  
  
#Web service query of sea surface temperature for NMFS Areas 640.  
  
head(httr::content(  
 httr::GET('https://apex.psmfc.org/akfin/data\_marts/akmp/nmfs\_area\_crw\_avg\_sst?nmfs\_area=640&start\_date=19850101&end\_date=20220101'),   
 type = "application/json") %>%   
 bind\_rows)

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| MEANSST | NMFSAREA | READ\_DATE | YEAR | JULIAN |
| 6.06 | 640 | 1985-01-01T12:00:00Z | 1985 | 001 |
| 5.89 | 640 | 1985-01-02T12:00:00Z | 1985 | 002 |
| 5.78 | 640 | 1985-01-03T12:00:00Z | 1985 | 003 |
| 5.76 | 640 | 1985-01-04T12:00:00Z | 1985 | 004 |
| 5.78 | 640 | 1985-01-05T12:00:00Z | 1985 | 005 |
| 5.92 | 640 | 1985-01-06T12:00:00Z | 1985 | 006 |

**R Code Input - 2**

#Web service query of sea surface temperature for NMFS Areas 640 and 650, averaged for June, July, and August  
  
httr::content(  
 httr::GET('https://apex.psmfc.org/akfin/data\_marts/akmp/nmfs\_area\_crw\_avg\_sst?nmfs\_area=640,650&start\_date=19850101&end\_date=20220101'),   
 type = "application/json") %>%   
 bind\_rows %>%   
 mutate(MONTH=month(as\_date(READ\_DATE))) %>% # Extract month   
 filter(MONTH==6 | MONTH==7 | MONTH==8) %>% # Filter summer months  
 group\_by(YEAR,NMFSAREA)%>%  
 summarize(SST=mean(MEANSST))%>% # Average by year and area.  
 ggplot(aes(as.numeric(YEAR),SST)) +   
 geom\_line() +   
 facet\_wrap(~NMFSAREA, nrow=2) +  
 xlab("Year") +  
 theme\_bw()



**R Code Input - 3**

#The full time series yields more than 13,000 rows of data per area (i.e., daily data from 1985-01-01 to present).   
  
data <- httr::content(  
 httr::GET('https://apex.psmfc.org/akfin/data\_marts/akmp/nmfs\_area\_crw\_avg\_sst?nmfs\_area=640&start\_date=19850101&end\_date=20220101'),   
 type = "application/json") %>%   
 bind\_rows  
  
str(data)

## tibble [13,348 x 5] (S3: tbl\_df/tbl/data.frame)  
## $ MEANSST : num [1:13348] 6.06 5.89 5.78 5.76 5.78 5.92 6 5.92 5.85 5.83 ...  
## $ NMFSAREA : chr [1:13348] "640" "640" "640" "640" ...  
## $ READ\_DATE: chr [1:13348] "1985-01-01T12:00:00Z" "1985-01-02T12:00:00Z" "1985-01-03T12:00:00Z" "1985-01-04T12:00:00Z" ...  
## $ YEAR : chr [1:13348] "1985" "1985" "1985" "1985" ...  
## $ JULIAN : chr [1:13348] "001" "002" "003" "004" ...

**R Code Input - 4**

#SST in NMFS area 640 in 1987.   
  
httr::content(  
 httr::GET('https://apex.psmfc.org/akfin/data\_marts/akmp/nmfs\_area\_crw\_avg\_sst?nmfs\_area=640&start\_date=19870101&end\_date=19880101'),   
 type = "application/json") %>%   
 bind\_rows %>%   
 mutate(date=as\_date(READ\_DATE)) %>%   
 ggplot(aes(date,MEANSST)) +   
 geom\_line()+  
 theme\_bw()



**R Code Input - 5**

#You can query a specific date with "read\_date". For example SST in MFS 640 on Y2K.  
  
#Query the day after your date of interest because omitting the time component in read\_date misses that day's reading.  
httr::content(  
 httr::GET('https://apex.psmfc.org/akfin/data\_marts/akmp/nmfs\_area\_crw\_avg\_sst?nmfs\_area=640&read\_date=20000101'),   
 type = "application/json") %>%   
 bind\_rows

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| MEANSST | NMFSAREA | READ\_DATE | YEAR | JULIAN |
| 4.94 | 640 | 1999-12-31T12:00:00Z | 1999 | 365 |

**R Code Input - 6**

#You can specify a number of days prior to any date using a "days\_back" parameter specification. For example the three days before Y2K.   
  
httr::content(  
 httr::GET('https://apex.psmfc.org/akfin/data\_marts/akmp/nmfs\_area\_crw\_avg\_sst?nmfs\_area=640&read\_date=20000101&days\_back=2'),   
 type = "application/json") %>%   
 bind\_rows

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| MEANSST | NMFSAREA | READ\_DATE | YEAR | JULIAN |
| 5.43 | 640 | 1999-12-29T12:00:00Z | 1999 | 363 |
| 5.13 | 640 | 1999-12-30T12:00:00Z | 1999 | 364 |
| 4.94 | 640 | 1999-12-31T12:00:00Z | 1999 | 365 |

**R Code Input - 7**

#If "read\_date" is not specified, "days\_back" returns the most recent SSTs. Here are SSTs for the last three days in NMFS 640.  
  
httr::content(  
 httr::GET('https://apex.psmfc.org/akfin/data\_marts/akmp/nmfs\_area\_crw\_avg\_sst?nmfs\_area=640&days\_back=2'),   
 type = "application/json") %>%   
 bind\_rows

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| MEANSST | NMFSAREA | READ\_DATE | YEAR | JULIAN |
| 11.86 | 640 | 2021-07-17T12:00:00Z | 2021 | 198 |
| 11.72 | 640 | 2021-07-18T12:00:00Z | 2021 | 199 |
| 12.28 | 640 | 2021-07-19T12:00:00Z | 2021 | 200 |

**R Code Input - 8**

#To query multiple areas, separate the values by a comma. For example to query NMFS areas 640 and 650 (Southeast Alaska outside waters).  
  
httr::content(  
 httr::GET("https://apex.psmfc.org/akfin/data\_marts/akmp/nmfs\_area\_crw\_avg\_sst?nmfs\_area=640,650"),   
 type = "application/json")%>%   
 bind\_rows

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| MEANSST | NMFSAREA | READ\_DATE | YEAR | JULIAN |
| 12.28 | 640 | 2021-07-19T12:00:00Z | 2021 | 200 |
| 13.22 | 650 | 2021-07-19T12:00:00Z | 2021 | 200 |

**R Code Input - 9**

#View strata included in the lookup table  
lkp <- readRDS("Data/crwsst\_spatial\_lookup\_table.RDS")   
  
unique(lkp$Ecosystem\_sub)

## [1] NA "Northern Bering Sea"   
## [3] "Western Gulf of Alaska" "Eastern Gulf of Alaska"   
## [5] "Southeastern Bering Sea" "Eastern Aleutians"   
## [7] "Central Aleutians" "Western Aleutians"

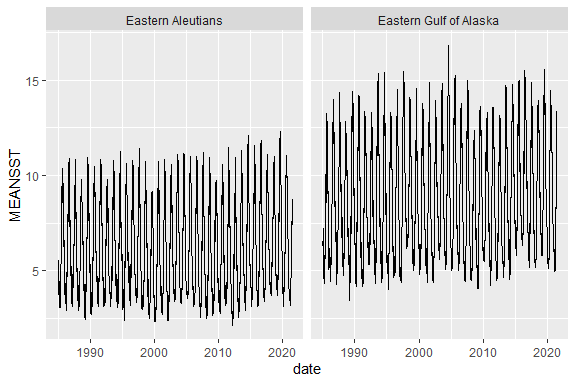
**R Code Input - 10**

#To query the data for the "Southeastern Bering Sea", for example, add "ecosystem\_sub=Southeastern%20Bering%20Sea", where spaces are filled by "%20".  
  
httr::content(  
 httr::GET('https://apex.psmfc.org/akfin/data\_marts/akmp/ecosystem\_sub\_crw\_avg\_sst?ecosystem\_sub=Southeastern%20Bering%20Sea'),   
 type = "application/json") %>%  
 bind\_rows

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| MEANSST | ECOSYSTEM\_SUB | READ\_DATE | YEAR | JULIAN |
| 9.39 | Southeastern Bering Sea | 2021-07-19T12:00:00Z | 2021 | 200 |

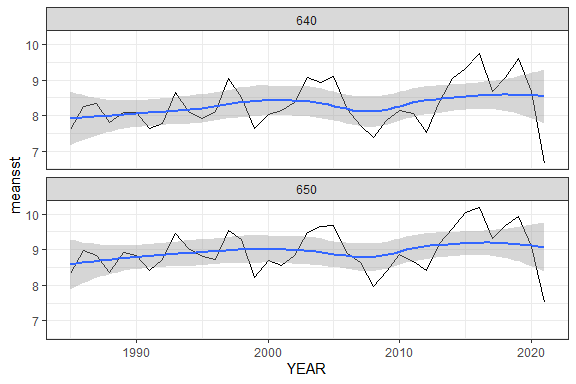
**R Code Input - 11**

#Eastern GOA and Eastern Aleutians SST from 1985 - Present."  
  
# This is Fig. 4 in the main body  
  
httr::content(  
 httr::GET('https://apex.psmfc.org/akfin/data\_marts/akmp/ecosystem\_sub\_crw\_avg\_sst?ecosystem\_sub=Eastern%20Gulf%20of%20Alaska,Eastern%20Aleutians&start\_date=19850101&end\_date=20220101'),   
 type = "application/json") %>%   
 bind\_rows %>%   
 mutate(date=as\_date(READ\_DATE)) %>%   
 ggplot(aes(date,MEANSST)) +   
 geom\_line() +   
 facet\_wrap(~ECOSYSTEM\_SUB)



**R Code Input - 12**

#"Annual average SST for NMFS areas 640 and 650."  
  
httr::content(  
 httr::GET('https://apex.psmfc.org/akfin/data\_marts/akmp/nmfs\_area\_crw\_avg\_sst?nmfs\_area=640,650&start\_date=19850101&end\_date=20220101'),   
 type = "application/json") %>%   
 bind\_rows %>%   
 mutate(date=as\_date(READ\_DATE),  
 YEAR=as.numeric(YEAR)) %>%   
 group\_by(YEAR,NMFSAREA) %>%   
 summarise(meansst=mean(MEANSST)) %>%   
 ggplot(aes(YEAR,meansst)) +   
 geom\_line() +   
 geom\_smooth() +  
 facet\_wrap(~NMFSAREA, nrow=2)+  
 theme\_bw()



**R Code Input - 13**

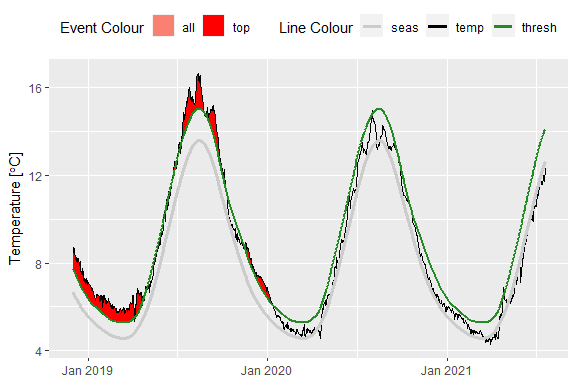
#Marine heatwave calculation (Schlegel et al. 2018) for NMFS region 640.  
#Step 1: Get SST data using httr  
  
updateddata <- httr::content(  
 httr::GET('https://apex.psmfc.org/akfin/data\_marts/akmp/nmfs\_area\_crw\_avg\_sst?nmfs\_area=640&start\_date=19850101&end\_date=20211231'),   
 type = "application/json") %>%   
 bind\_rows %>%   
 mutate(date=as\_date(READ\_DATE)) %>%   
 data.frame %>%   
 dplyr::select(date,  
 meansst=MEANSST,  
 NMFSAREA) #simplify data frame for clarity.

**R Code Input - 14**

#Step 2: Calculate marine heatwave indices using the mhw package.  
library(heatwaveR)  
mhw <- detect\_event(  
 ts2clm(updateddata %>%   
 rename(t=date,  
 temp=meansst) %>%   
 arrange(t),  
 climatologyPeriod = c("1985-01-01", "2014-12-31"))) #Specify baseline period

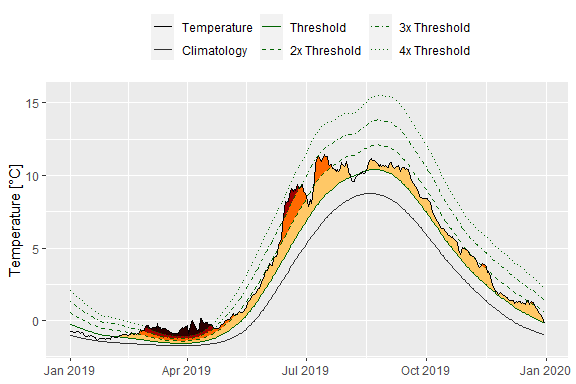
**R Code Input - 15**

#Create "flame" plots.  
# Plotting code directly from heatwaveR vignette.  
  
mhw\_clim <- mhw$climatology %>%   
 filter(t>=as.Date("2018-12-01")) #Extract the MHW data  
  
ggplot(data = mhw\_clim,   
 aes(x = t)) +  
 geom\_flame(aes(y = temp, y2 = thresh, fill = "all"),  
 show.legend = T) +  
 geom\_flame(data = mhw\_clim,   
 aes(y = temp,y2 = thresh, fill = "top"),   
 show.legend = T) +  
 geom\_line(aes(y = temp, colour = "temp")) +  
 geom\_line(aes(y = thresh, colour = "thresh"), size = 1.0) +  
 geom\_line(aes(y = seas, colour = "seas"), size = 1.2) +  
 scale\_colour\_manual(name = "Line Colour",  
 values = c("temp" = "black",   
 "thresh" = "forestgreen",   
 "seas" = "grey80")) +  
 scale\_fill\_manual(name = "Event Colour",   
 values = c("all" = "salmon",   
 "top" = "red")) +  
 scale\_x\_date(date\_labels = "%b %Y") +  
 guides(colour = guide\_legend(  
 override.aes = list(fill = NA))) +  
 labs(y = expression(paste("Temperature [", degree, "C]")), x = NULL) +   
 theme(legend.position="top")



**R Code Input - 16**

# A better illustration of heatwave categories from the Northern Bering Sea  
  
# Here we'll use an example where we do not save the SST data as a separate object first, simply embedding it into the ts2clm() function.  
  
#This is figure 5 in the main body.  
  
clim\_cat <- (detect\_event(  
 ts2clm(  
 httr::content(  
 httr::GET('https://apex.psmfc.org/akfin/data\_marts/akmp/ecosystem\_sub\_crw\_avg\_sst?ecosystem\_sub=Northern%20Bering%20Sea&start\_date=19850101&end\_date=20211231'),  
 type = "application/json") %>%   
 bind\_rows %>%   
 mutate(date=as\_date(READ\_DATE)) %>%   
 data.frame %>%   
 dplyr::select(t=date,temp=MEANSST) %>%   
 arrange(t),   
 climatologyPeriod = c("1985-01-01", "2014-12-31"))))$climatology %>% #Specify the baseline period.  
 dplyr::mutate(diff = thresh - seas,  
 thresh\_2x = thresh + diff,  
 thresh\_3x = thresh\_2x + diff,  
 thresh\_4x = thresh\_3x + diff) %>%   
 filter(t>=as.Date("2019-01-01") & t<=as.Date("2019-12-31")) # Select the time period to display.  
  
# Plotting code directly from heatwaveR vignette.  
# Set line colours  
lineColCat <- c(  
 "Temperature" = "black",  
 "Climatology" = "gray20",  
 "Threshold" = "darkgreen",  
 "2x Threshold" = "darkgreen",  
 "3x Threshold" = "darkgreen",  
 "4x Threshold" = "darkgreen"  
)  
  
# Set category fill colours  
fillColCat <- c(  
 "Moderate" = "#ffc866",  
 "Strong" = "#ff6900",  
 "Severe" = "#9e0000",  
 "Extreme" = "#2d0000"  
)  
  
ggplot(data = clim\_cat,   
 aes(x = t, y = temp)) +  
 geom\_flame(aes(y2 = thresh, fill = "Moderate")) +  
 geom\_flame(aes(y2 = thresh\_2x, fill = "Strong")) +  
 geom\_flame(aes(y2 = thresh\_3x, fill = "Severe")) +  
 geom\_flame(aes(y2 = thresh\_4x, fill = "Extreme")) +  
 geom\_line(aes(y = thresh\_2x, col = "2x Threshold"),  
 size = 0.7, linetype = "dashed") +  
 geom\_line(aes(y = thresh\_3x, col = "3x Threshold"),   
 size = 0.7, linetype = "dotdash") +  
 geom\_line(aes(y = thresh\_4x, col = "4x Threshold"),   
 size = 0.7, linetype = "dotted") +  
 geom\_line(aes(y = seas,col = "Climatology"), size = 0.7) +  
 geom\_line(aes(y = thresh,col = "Threshold"), size = 0.7) +  
 geom\_line(aes(y = temp,col = "Temperature"), size = 0.6) +  
 scale\_colour\_manual(name = NULL,   
 values = lineColCat,  
 breaks = c("Temperature",   
 "Climatology",  
 "Threshold",  
 "2x Threshold",   
 "3x Threshold",  
 "4x Threshold")) +  
 scale\_fill\_manual(name = NULL,   
 values = fillColCat,   
 guide = FALSE) +  
 scale\_x\_date(date\_labels = "%b %Y") +  
 guides(colour = guide\_legend(  
 override.aes = list(linetype = c("solid",   
 "solid",   
 "solid",  
 "dashed",   
 "dotdash",   
 "dotted"),  
 size = c(0.6, 0.7, 0.7,0.7, 0.7, 0.7)))) +  
 labs(y = "Temperature [°C]", x = NULL) +  
 theme(legend.position="top")



**Oracle database queries**

**R Code Input - 17**

#Connect to the AKFIN database with R  
  
# Load the AKFIN database user name and password from an external file.  
params <- read\_csv("markdown\_odbc\_params.csv")  
  
# Connect to the AKFIN database  
con <- dbConnect(odbc::odbc(), "akfin", UID=params$uid, PWD=params$pass)

**SQL Code Input - 1**

-- Spatial lookup table on the AKFIN Oracle database  
  
select \* from afsc.erddap\_crw\_sst\_spatial\_lookup   
where rownum<=5

5 records

|  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| ID | STATE\_FED | STAT\_AREA | DEPTH | LONGITUDE | LATITUDE | NMFSAREA | BSIERP\_ID | BSIERP\_NAME | ECOSYSTEM | ECOSYSTEM\_SUB | CRAB |
| 58040 | FED | 525902 | -126 | -152.175 | 59.175 | 630 | NA | NA | Gulf of Alaska | Western Gulf of Alaska | NA |
| 58041 | FED | 525902 | -130 | -152.125 | 59.175 | 630 | NA | NA | Gulf of Alaska | Western Gulf of Alaska | NA |
| 58042 | FED | 525902 | -119 | -152.075 | 59.175 | 630 | NA | NA | Gulf of Alaska | Western Gulf of Alaska | NA |
| 58043 | FED | 525902 | -108 | -152.025 | 59.175 | 630 | NA | NA | Gulf of Alaska | Western Gulf of Alaska | NA |
| 58044 | STATE | 515906 | -115 | -151.975 | 59.175 | 630 | NA | NA | Gulf of Alaska | Western Gulf of Alaska | NA |

**SQL Code Input - 2**

-- CRW SST from the AKFIN Oracle database  
  
select \* from afsc.erddap\_crw\_sst   
where rownum<=5

5 records

|  |  |  |
| --- | --- | --- |
| READ\_DATE | CRW\_ID | TEMP |
| 1985-04-03 12:00:00 | 66728 | -1.72 |
| 1985-04-03 12:00:00 | 66729 | -1.72 |
| 1985-04-03 12:00:00 | 66730 | -1.71 |
| 1985-04-03 12:00:00 | 66731 | -1.70 |
| 1985-04-03 12:00:00 | 66732 | -1.69 |

**SQL Code Input - 3**

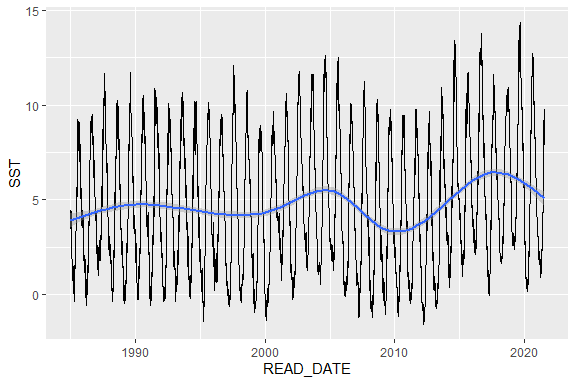
-- query SST (“TEMP”) data that fall within a crab management area and add a field for “Year”  
  
select read\_date,  
 temp,  
 to\_char(read\_date,'YYYY') as Year,  
 crab  
from afsc.erddap\_crw\_sst a  
INNER JOIN (select \* from afsc.erddap\_crw\_sst\_spatial\_lookup  
where crab <> 'NA') b  
ON a.crw\_id =b.id  
where rownum<=5

5 records

|  |  |  |  |
| --- | --- | --- | --- |
| READ\_DATE | TEMP | YEAR | CRAB |
| 2007-01-05 12:00:00 | 1.58 | 2007 | stmatts |
| 2007-01-06 12:00:00 | 1.29 | 2007 | stmatts |
| 2007-01-07 12:00:00 | 1.35 | 2007 | stmatts |
| 2007-01-08 12:00:00 | 1.62 | 2007 | stmatts |
| 2007-01-13 12:00:00 | 1.58 | 2007 | stmatts |

**R / SQL Code Input - 1**

#Plotted query of Bristol Bay crab management area SST data averaged daily and plotted with default smoothing.  
  
dbFetch(dbSendQuery(con,  
 paste0("select read\_date,  
 round(avg(temp),2) as sst,  
 crab  
 from afsc.erddap\_crw\_sst a  
 INNER JOIN (select \*   
 from afsc.erddap\_crw\_sst\_spatial\_lookup  
 where crab = 'bb') b  
 ON a.crw\_id = b.id   
 group by   
 crab,  
 read\_date"))) %>%   
 ggplot(aes(READ\_DATE,SST)) +   
 geom\_line() +   
 geom\_smooth()



# References

Barbeaux SJ, Holsman K, Zador S. 2020. Marine Heatwave Stress Test of Ecosystem-Based Fisheries Management in the Gulf of Alaska Pacific Cod Fishery. Frontiers in Marine Science 7: 1–21.

Fredston A, Pinsky M, Selden RL, Szuwalski C, Thorson JT, Gaines SD, Halpern BS. 2021. Range edges of North American marine species are tracking temperature over decades. Global Change Biology 1–12.

Gaichas SK, Link JS, Hare JA. 2014. A risk-based approach to evaluating northeast US fish community vulnerability to climate change. ICES Journal of Marine Science 71: 2323–2342.

Harley JR, Lanphier K, Kennedy E, Whitehead C, Bidlack A. 2020. Random forest classification to determine environmental drivers and forecast paralytic shellfish toxins in Southeast Alaska with high temporal resolution. Harmful Algae 99: 101918.

Hazen EL, Scales KL, Maxwell SM, Briscoe DK, Welch H, Bograd SJ, Bailey H, Benson SR, Eguchi T, Dewar H, Kohin S, Costa DP, Crowder LB, Lewison RL. 2018. A dynamic ocean management tool to reduce bycatch and support sustainable fisheries. Science Advances 4: eaar3001.

Holsman KK, Ianelli J, Aydin K, Punt AE, Moffitt EA. 2016. A comparison of fisheries biological reference points estimated from temperature-specific multi-species and single-species climate-enhanced stock assessment models. Deep-Sea Research Part II: Topical Studies in Oceanography 134: 360–378.

Kotwicki S, Lauth RR. 2013. Detecting temporal trends and environmentally-driven changes in the spatial distribution of bottom fishes and crabs on the eastern Bering Sea shelf. Deep-Sea Research Part II: Topical Studies in Oceanography 94: 231–243.

Pinsky ML, Rogers LA, Morley JW, Frölicher TL. 2020. Ocean planning for species on the move provides substantial benefits and requires few trade-offs. Science Advances 6:

Rogers LA, Griffin R, Young T, Fuller E, Martin KS, Pinsky ML. 2019. Shifting habitats expose fishing communities to risk under climate change. Nature Climate Change 9: 512–516.

Simons B. 2020. ERDDAP.