

Results: 20, 30, or 60

datetime YYYY-mm-dd hr hr:min min:sec sec -08

datetime 2-1 2008-06-20 15:00:00

2008-06-20 15:30:00

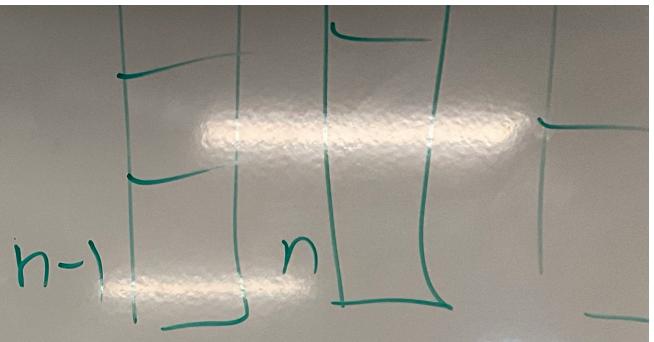
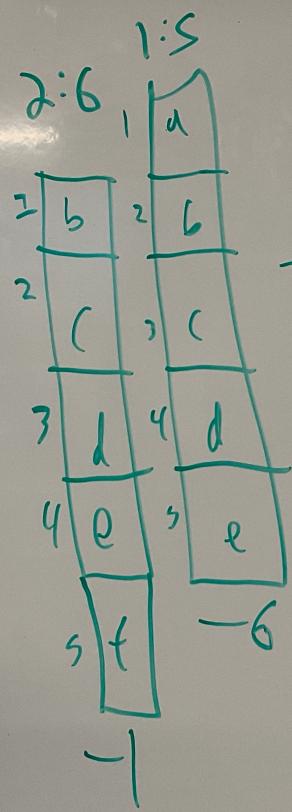
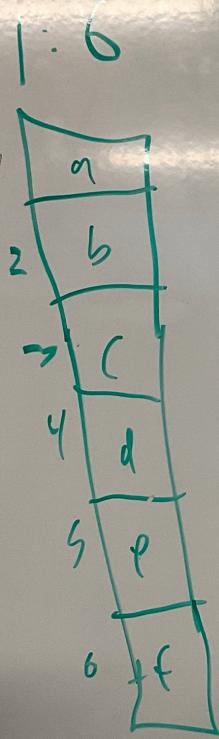
⇒ time difference of 30 minutes,

harris_interval: datetime 2-1, 3-2 ... 2187966-2187965

~~harris_interval ← datetime[-1] - datetime[-2]~~

time difference

I) harris_datetime contains only sampling events from harris sites



-Interval

$$1 = b - a$$

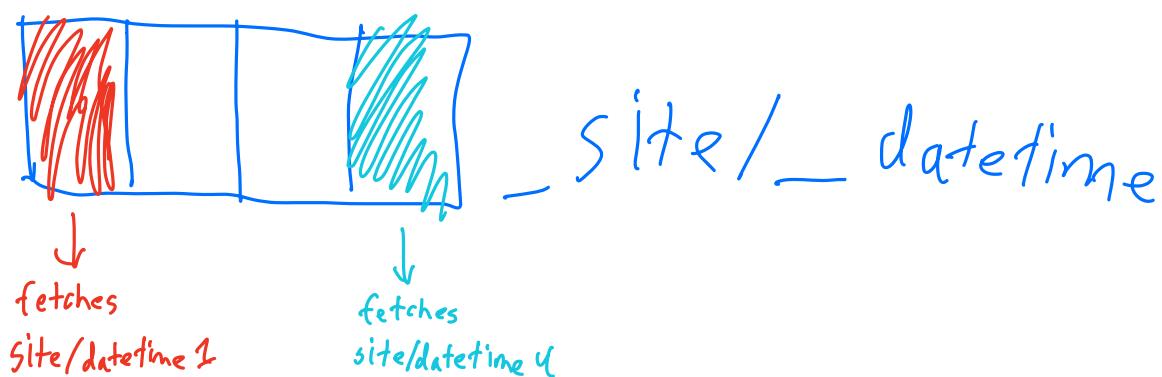
$$2 = c - b$$

$$3 = d - c$$

$$4 = e - d$$

$$5 = f - e$$

- ① store the output of `which.max(_temp)` as a variable. `hottest_idx` now directs to element 158962 when used to index.
- ② use that variable to store the hottest time and site



$$-\text{site}[\text{which.min}(-\text{temperature})] = -\text{site}[4]$$

A familiar way of thinking about it: $f(g(x))$

hottest=variables

- idx: element #
- time: time
- site: site name
- time_obj: time as an obj for start & end vars
- start: 12 am of day with hottest measurement
- end: 12 am of next day

Hotday=rectors

Hotday-idx: TRUE FALSE for all elements in our vectors

Vector visualization: Table

_datetime						
-temp						
-exposure						
:						

hottest.idx is like selecting a row

hotday.idx is like selecting several rows

each column is all the data/meta data for 1 sampling event

.idx → index var/vectors that are used in vector brackets

* V* means vector or variable

.idx

Variable ← function that selects 1 element of a vector → stores the element # of the element selected by the function
↳ variable containing an element of type integer

Vector₂ ← vector w/condition(s) → stores the element # where that condition(s) is met. → vector containing 'TRUE' or 'False' for every element in the original

V* ← vector [#/idx/condition] → stores elements of the original vector. → V* contains elements of the same type as the original vector → e.g. decimal #, string, date time, etc.

Final step: average # of hours extreme-summer/summer length

* # of days to be averaging over?

of extreme hours

extreme-summer \leftarrow extreme-summer/2 convert # of 30 mins into # of hours

extreme_Summer \leftarrow length(summer_extimes) Number of extremes at Muko Bay in the summer

summer_ex-times : `ketj_datetime[summer_ex_idx]` pulls out every sampling event with extreme temps at Nuka Bay in the summer Try combining these steps by doing `~datetime[!idx, -idx]`

summer_ex_idx: in summer & at Nuka bay & extreme temp Make an index for extreme sampling events
only at Nuka bay in the summer

`extreme_temp_idx: ≥ 25 or ≤ -4` Make an index for everywhere there's an extreme air
(kef_temperature ≥ 25 | kef_temperature ≤ -4) temp → this can be reused
for the next site/season
Recording air temp
(exposure == "air" | exposure == "air/transition") ↗ not technically necessary for this data set b/c only
air temp are variable enough to meet the first condition anyways

of days in the season at the site

`summer_idx`: index in summer & at Nukabay. Index for every time both conditions are met
`_season == summer` `_site == Nukabay`

			Select only row 3 b/c both conditions are met	
Season	Summer	Fall	Summer	Spring
site	Harris	Aialik	Nuka bay	Nuka bay

summer times: `ketj_datetime[summer_idx]` pulls out every sampling event at Nuka Bay in the summer These two steps could be replaced with: `summerlength < length(summer_idx == TRUE)`

Summer-length: length(summer-times) Actual times don't matter, just the # of times there are in
but I like this more it's clearer

summer_length: summer_length / 48 turns # of 30 min \rightarrow # of days

1 site
 mean of $k_{efj,temp}$
 ↗ season 1 year ("mean seasonal water temp" seems to refer to seasons mean for a given year in Traiger et al. (2012))
Subtracting logger Mean seasonal water temperatures from the Mean₂
 (regional mean) for the whole time series (within each of the five
 time periods.) → only 1 season at a time
 ↗ Just when the logger was recording
 water temp → exposure is "water"
 mean of $k_{efj,temp}$
 all sites
 included

	Mean ₁	Mean ₂
sites	All k _{efj}	Just 1 site
years	All years w/data	Just 1 year → no idea how to just scoop up the data for 2
seasons	Just 1	Year... I've tried.
exposure	Just water	Just Water

Get the temperature anomalies for one site in one season in one year

`kefj_spring: kefj_temp[kefj_season == 'spring' & -exposure == 'water']`

`harris_spring: kefj_temp[kefj_season == 'spring' &
-exposure == 'water' &
-site == 'Harris']`

Year is 2014 `Not_datetime == 2018 or 2018*`
`or '2018' or '2018**'`

↳ 2. format_datetime so R studio recognizes
Is there a the elements as times.

Simpler way to do 2. Extract the years - make a vector that is
this? Perhaps, but just the year for every sampling event.

this is what google 3. Make an index vector that is true only
had for me. when the year vector is 2014

last step: $(\text{mean}(\text{kefj_spring})) - (\text{mean}(\text{Harris_spring}))$