Satellite image comparison

```
library(tidyverse)
```

```
## -- Attaching packages --------------------------------------- tidyverse 1.3.1 --
```

```
## v ggplot2 3.3.5          v purrr   0.3.4
## v tibble  3.1.5          v dplyr   1.0.7
## v tidyr   1.1.4          v stringr 1.4.0.9000
## v readr   2.0.2          v forcats 0.5.1
```

```
## -- Conflicts ------------------------------------------ tidyverse_conflicts() --
## x dplyr::filter() masks stats::filter()
## x dplyr::lag()    masks stats::lag()
```

```
library(here)
```

```
## here() starts at /home/cengstro/ownCloud/proj/radiometer
```

```
library(fs)
library(janitor)
```

```
##
## Attaching package: 'janitor'
```

```
## The following objects are masked from 'package:stats':
##
##     chisq.test, fisher.test
```

```
library(lubridate)
```

```
##
## Attaching package: 'lubridate'
```

```
## The following objects are masked from 'package:base':
##
##     date, intersect, setdiff, union
```

```
library(zoo)
```

```
##
## Attaching package: 'zoo'
```

```
## The following objects are masked from 'package:base':
##
##     as.Date, as.Date.numeric
```

```
library(kableExtra)
```

```
##
## Attaching package: 'kableExtra'
```

```
## The following object is masked from 'package:dplyr':
##
##     group_rows
```

```r
library(GGally)
```

```
## Registered S3 method overwritten by 'GGally':
##   method from
##   +.gg   ggplot2
```

```r
ts_modis <- dir_ls(here("data/gee_ts/modis_grid_cells"), regexp = ".csv$") %>%
  map_df(read_csv, id = "filename")
```

```
## Rows: 17 Columns: 3

## -- Column specification --------------------------------------------------------
## Delimiter: ","
## chr (1): system:time_start
## dbl (1): RGND_CLOUDMASKED

##
## i Use `spec()` to retrieve the full column specification for this data.
## i Specify the column types or set `show_col_types = FALSE` to quiet this message.

## Rows: 38 Columns: 3

## -- Column specification --------------------------------------------------------
## Delimiter: ","
## chr (1): system:time_start
## dbl (1): ALBEDO

##
## i Use `spec()` to retrieve the full column specification for this data.
## i Specify the column types or set `show_col_types = FALSE` to quiet this message.

## Rows: 3 Columns: 3

## -- Column specification --------------------------------------------------------
## Delimiter: ","
## chr (1): system:time_start
## dbl (1): RGND

##
## i Use `spec()` to retrieve the full column specification for this data.
## i Specify the column types or set `show_col_types = FALSE` to quiet this message.

## Rows: 13 Columns: 3

## -- Column specification --------------------------------------------------------
## Delimiter: ","
## chr (1): system:time_start
## dbl (1): RGND_CLOUDMASKED

##
## i Use `spec()` to retrieve the full column specification for this data.
## i Specify the column types or set `show_col_types = FALSE` to quiet this message.

## Rows: 34 Columns: 3

## -- Column specification --------------------------------------------------------
## Delimiter: ","
## chr (1): system:time_start
## dbl (1): RGND_CLOUDMASKED

##
```

```
## i Use `spec()` to retrieve the full column specification for this data.
## i Specify the column types or set `show_col_types = FALSE` to quiet this message.

## Rows: 44 Columns: 3

## -- Column specification --------------------------------------------------------
## Delimiter: ","
## chr (1): system:time_start
## dbl (1): ALBEDO

##
## i Use `spec()` to retrieve the full column specification for this data.
## i Specify the column types or set `show_col_types = FALSE` to quiet this message.

## Rows: 17 Columns: 3

## -- Column specification --------------------------------------------------------
## Delimiter: ","
## chr (1): system:time_start
## dbl (1): RGND_CLOUDMASKED

##
## i Use `spec()` to retrieve the full column specification for this data.
## i Specify the column types or set `show_col_types = FALSE` to quiet this message.

## Rows: 28 Columns: 3

## -- Column specification --------------------------------------------------------
## Delimiter: ","
## chr (1): system:time_start
## dbl (1): ALBEDO

##
## i Use `spec()` to retrieve the full column specification for this data.
## i Specify the column types or set `show_col_types = FALSE` to quiet this message.

## Rows: 4 Columns: 3

## -- Column specification --------------------------------------------------------
## Delimiter: ","
## chr (1): system:time_start
## dbl (1): RGND

##
## i Use `spec()` to retrieve the full column specification for this data.
## i Specify the column types or set `show_col_types = FALSE` to quiet this message.

## Rows: 8 Columns: 3

## -- Column specification --------------------------------------------------------
## Delimiter: ","
## chr (1): system:time_start
## dbl (1): RGND_CLOUDMASKED

##
## i Use `spec()` to retrieve the full column specification for this data.
## i Specify the column types or set `show_col_types = FALSE` to quiet this message.

## Rows: 30 Columns: 3

## -- Column specification --------------------------------------------------------
## Delimiter: ","
## chr (1): system:time_start
## dbl (1): RGND_CLOUDMASKED
```

```
##
## i Use `spec()` to retrieve the full column specification for this data.
## i Specify the column types or set `show_col_types = FALSE` to quiet this message.
```

```
## Rows: 41 Columns: 3
```

```
## -- Column specification --------------------------------------------------------
## Delimiter: ","
## chr (1): system:time_start
## dbl (1): ALBEDO
```

```
##
## i Use `spec()` to retrieve the full column specification for this data.
## i Specify the column types or set `show_col_types = FALSE` to quiet this message.
```

```
# w custom perimeter
ts_custom <- dir_ls(here("data/gee_ts/custom_polygon"), regexp = ".csv$") %>%
  map_df(read_csv, id = "filename")
```

```
## Rows: 17 Columns: 3
```

```
## -- Column specification --------------------------------------------------------
## Delimiter: ","
## chr (1): system:time_start
## dbl (1): RGND_CLOUDMASKED
```

```
##
## i Use `spec()` to retrieve the full column specification for this data.
## i Specify the column types or set `show_col_types = FALSE` to quiet this message.
```

```
## Rows: 38 Columns: 3
```

```
## -- Column specification --------------------------------------------------------
## Delimiter: ","
## chr (1): system:time_start
## dbl (1): ALBEDO
```

```
##
## i Use `spec()` to retrieve the full column specification for this data.
## i Specify the column types or set `show_col_types = FALSE` to quiet this message.
```

```
## Rows: 3 Columns: 3
```

```
## -- Column specification --------------------------------------------------------
## Delimiter: ","
## chr (1): system:time_start
## dbl (1): RGND
```

```
##
## i Use `spec()` to retrieve the full column specification for this data.
## i Specify the column types or set `show_col_types = FALSE` to quiet this message.
```

```
## Rows: 13 Columns: 3
```

```
## -- Column specification --------------------------------------------------------
## Delimiter: ","
## chr (1): system:time_start
## dbl (1): RGND_CLOUDMASKED
```

```
##
## i Use `spec()` to retrieve the full column specification for this data.
## i Specify the column types or set `show_col_types = FALSE` to quiet this message.

## Rows: 34 Columns: 3

## -- Column specification --------------------------------------------------------
## Delimiter: ","
## chr (1): system:time_start
## dbl (1): RGND_CLOUDMASKED

##
## i Use `spec()` to retrieve the full column specification for this data.
## i Specify the column types or set `show_col_types = FALSE` to quiet this message.

## Rows: 44 Columns: 3

## -- Column specification --------------------------------------------------------
## Delimiter: ","
## chr (1): system:time_start
## dbl (1): ALBEDO

##
## i Use `spec()` to retrieve the full column specification for this data.
## i Specify the column types or set `show_col_types = FALSE` to quiet this message.

## Rows: 17 Columns: 3

## -- Column specification --------------------------------------------------------
## Delimiter: ","
## chr (1): system:time_start
## dbl (1): RGND_CLOUDMASKED

##
## i Use `spec()` to retrieve the full column specification for this data.
## i Specify the column types or set `show_col_types = FALSE` to quiet this message.

## Rows: 28 Columns: 3

## -- Column specification --------------------------------------------------------
## Delimiter: ","
## chr (1): system:time_start
## dbl (1): ALBEDO

##
## i Use `spec()` to retrieve the full column specification for this data.
## i Specify the column types or set `show_col_types = FALSE` to quiet this message.

## Rows: 4 Columns: 3

## -- Column specification --------------------------------------------------------
## Delimiter: ","
## chr (1): system:time_start
## dbl (1): RGND

##
## i Use `spec()` to retrieve the full column specification for this data.
## i Specify the column types or set `show_col_types = FALSE` to quiet this message.

## Rows: 8 Columns: 3

## -- Column specification --------------------------------------------------------
## Delimiter: ","
## chr (1): system:time_start
## dbl (1): RGND_CLOUDMASKED

##
## i Use `spec()` to retrieve the full column specification for this data.
## i Specify the column types or set `show_col_types = FALSE` to quiet this message.

## Rows: 30 Columns: 3

## -- Column specification --------------------------------------------------------
## Delimiter: ","
## chr (1): system:time_start
## dbl (1): RGND_CLOUDMASKED

##
## i Use `spec()` to retrieve the full column specification for this data.
## i Specify the column types or set `show_col_types = FALSE` to quiet this message.

## Rows: 41 Columns: 3

## -- Column specification --------------------------------------------------------
## Delimiter: ","
## chr (1): system:time_start
## dbl (1): ALBEDO

##
## i Use `spec()` to retrieve the full column specification for this data.
## i Specify the column types or set `show_col_types = FALSE` to quiet this message.
```

```r
ts_glims <- dir_ls(here("data/gee_ts/glimsGeom"), regexp = ".csv$") %>%
  map_df(read_csv, id = "filename")
```

```
## Rows: 3 Columns: 3

## -- Column specification --------------------------------------------------------
## Delimiter: ","
## chr (1): system:time_start
## dbl (1): RGND

##
## i Use `spec()` to retrieve the full column specification for this data.
## i Specify the column types or set `show_col_types = FALSE` to quiet this message.

## Rows: 13 Columns: 3

## -- Column specification --------------------------------------------------------
## Delimiter: ","
## chr (1): system:time_start
## dbl (1): RGND_CLOUDMASKED

##
## i Use `spec()` to retrieve the full column specification for this data.
## i Specify the column types or set `show_col_types = FALSE` to quiet this message.

## Rows: 34 Columns: 3

## -- Column specification --------------------------------------------------------
## Delimiter: ","
## chr (1): system:time_start
## dbl (1): RGND_CLOUDMASKED
```

```
##
## i Use `spec()` to retrieve the full column specification for this data.
## i Specify the column types or set `show_col_types = FALSE` to quiet this message.

## Rows: 44 Columns: 3

## -- Column specification --------------------------------------------------------
## Delimiter: ","
## chr (1): system:time_start
## dbl (1): ALBEDO

##
## i Use `spec()` to retrieve the full column specification for this data.
## i Specify the column types or set `show_col_types = FALSE` to quiet this message.

## Rows: 4 Columns: 3

## -- Column specification --------------------------------------------------------
## Delimiter: ","
## chr (1): system:time_start
## dbl (1): RGND

##
## i Use `spec()` to retrieve the full column specification for this data.
## i Specify the column types or set `show_col_types = FALSE` to quiet this message.

## Rows: 8 Columns: 3

## -- Column specification --------------------------------------------------------
## Delimiter: ","
## chr (1): system:time_start
## dbl (1): RGND_CLOUDMASKED

##
## i Use `spec()` to retrieve the full column specification for this data.
## i Specify the column types or set `show_col_types = FALSE` to quiet this message.

## Rows: 30 Columns: 3

## -- Column specification --------------------------------------------------------
## Delimiter: ","
## chr (1): system:time_start
## dbl (1): RGND_CLOUDMASKED

##
## i Use `spec()` to retrieve the full column specification for this data.
## i Specify the column types or set `show_col_types = FALSE` to quiet this message.

## Rows: 41 Columns: 3

## -- Column specification --------------------------------------------------------
## Delimiter: ","
## chr (1): system:time_start
## dbl (1): ALBEDO

##
## i Use `spec()` to retrieve the full column specification for this data.
## i Specify the column types or set `show_col_types = FALSE` to quiet this message.
```

```
srad <- dir_ls(here("data"), regexp = "*srad_wx.csv$") %>% # doing this seperately for now, redundant s
  map_df(read_csv, id = "filename")
```

```
## Rows: 122 Columns: 3

## -- Column specification --------------------------------------------------------
## Delimiter: ","
## chr (1): system:time_start
## dbl (1): srad

##
## i Use `spec()` to retrieve the full column specification for this data.
## i Specify the column types or set `show_col_types = FALSE` to quiet this message.

## Rows: 122 Columns: 3

## -- Column specification --------------------------------------------------------
## Delimiter: ","
## chr (1): system:time_start
## dbl (1): srad

##
## i Use `spec()` to retrieve the full column specification for this data.
## i Specify the column types or set `show_col_types = FALSE` to quiet this message.
```

```
hist <- dir_ls(here("data"), regexp = "*hist.csv$") %>%
  map_df(read_csv, id = "filename")
```

```
## Rows: 254 Columns: 3

## -- Column specification --------------------------------------------------------
## Delimiter: ","
## dbl (2): Band Value, RGND Count

##
## i Use `spec()` to retrieve the full column specification for this data.
## i Specify the column types or set `show_col_types = FALSE` to quiet this message.

## Rows: 141 Columns: 3

## -- Column specification --------------------------------------------------------
## Delimiter: ","
## dbl (2): Band Value, RGND Count

##
## i Use `spec()` to retrieve the full column specification for this data.
## i Specify the column types or set `show_col_types = FALSE` to quiet this message.
```

```
# wx
wx <- dir_ls(here("data"), regexp = "*wx.csv$") %>%
  map_df(read_csv, id = "filename")
```

```
## Rows: 122 Columns: 3

## -- Column specification --------------------------------------------------------
## Delimiter: ","
## chr (1): system:time_start
## dbl (1): prcp

##
## i Use `spec()` to retrieve the full column specification for this data.
## i Specify the column types or set `show_col_types = FALSE` to quiet this message.

## Rows: 122 Columns: 3
## -- Column specification --------------------------------------------------------
## Delimiter: ","
## chr (1): system:time_start
## dbl (1): srad

##
## i Use `spec()` to retrieve the full column specification for this data.
## i Specify the column types or set `show_col_types = FALSE` to quiet this message.

## Rows: 122 Columns: 3

## -- Column specification --------------------------------------------------------
## Delimiter: ","
## chr (1): system:time_start
## dbl (1): tmax

##
## i Use `spec()` to retrieve the full column specification for this data.
## i Specify the column types or set `show_col_types = FALSE` to quiet this message.

## Rows: 122 Columns: 3

## -- Column specification --------------------------------------------------------
## Delimiter: ","
## chr (1): system:time_start
## dbl (1): tmin

##
## i Use `spec()` to retrieve the full column specification for this data.
## i Specify the column types or set `show_col_types = FALSE` to quiet this message.

## Rows: 122 Columns: 3

## -- Column specification --------------------------------------------------------
## Delimiter: ","
## chr (1): system:time_start
## dbl (1): prcp

##
## i Use `spec()` to retrieve the full column specification for this data.
## i Specify the column types or set `show_col_types = FALSE` to quiet this message.

## Rows: 122 Columns: 3

## -- Column specification --------------------------------------------------------
## Delimiter: ","
## chr (1): system:time_start
## dbl (1): srad

##
## i Use `spec()` to retrieve the full column specification for this data.
## i Specify the column types or set `show_col_types = FALSE` to quiet this message.

## Rows: 122 Columns: 3

## -- Column specification --------------------------------------------------------
## Delimiter: ","
## chr (1): system:time_start
## dbl (1): tmax

##
## i Use `spec()` to retrieve the full column specification for this data.
## i Specify the column types or set `show_col_types = FALSE` to quiet this message.
```

```
## Rows: 122 Columns: 3
```

```
## -- Column specification --------------------------------------------------------
## Delimiter: ","
## chr (1): system:time_start
## dbl (1): tmin
```

```
##
## i Use `spec()` to retrieve the full column specification for this data.
## i Specify the column types or set `show_col_types = FALSE` to quiet this message.
```

tidy

```
# parse site name, date
my_tidy <- function(df){
  df %>%
    clean_names() %>%
    mutate(site = filename %>% basename() %>% str_remove(".csv") %>% str_split_n("_", 1),
           date = mdy(system_time_start))
}

# parse platform name, apply cloudmask
ts_tidy <- function(df){
  df %>%
     mutate(platform = basename(filename) %>%
              str_split_n("_", 2) %>%
              str_remove(".csv"),
         rgnd = if_else(is.na(rgnd), rgnd_cloudmasked, rgnd)) %>%
  select(platform, site, date, rgnd, albedo)
}


ts_modis_wide <- ts_modis %>%
  my_tidy() %>%
  ts_tidy()

ts_custom_wide <- ts_custom %>%
  my_tidy() %>%
  ts_tidy()

ts_glims_wide <- ts_glims %>%
  my_tidy() %>%
  ts_tidy()

srad_clean <- srad %>%
  my_tidy() %>%
  select(site, date, srad)

hist_clean <- hist %>%
  clean_names() %>%
  mutate(site = basename(filename) %>% str_split_n("_", 1), .keep = "unused")
```

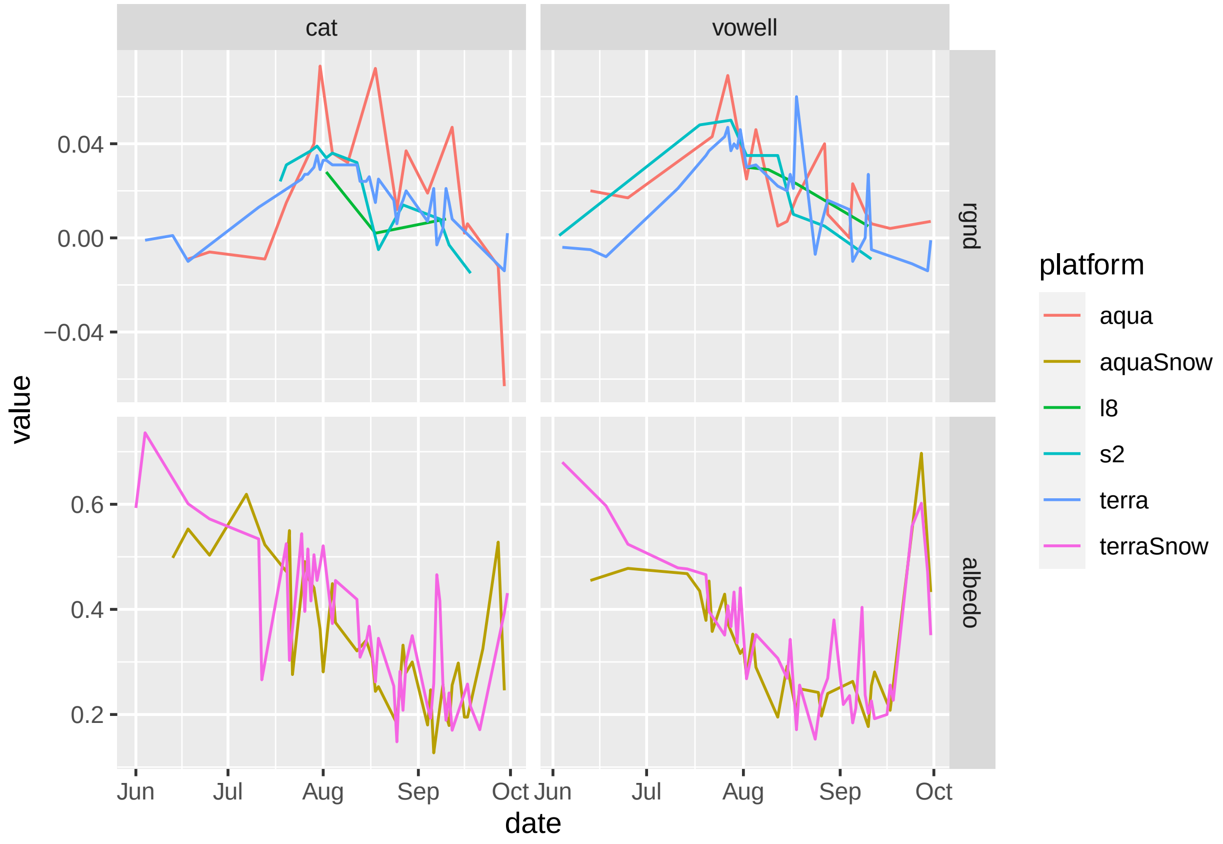
```
wx_clean <- wx %>%
  my_tidy() %>%
  select(-filename, -system_time_start) %>%
  pivot_longer(-site:-date, values_drop_na =TRUE) %>%
  pivot_wider()
```

Long format for plotting w albedo

```
# for plotting albedo and RGND on the same axis
ts_modis_long <- ts_modis_wide %>%
  pivot_longer(cols = c(rgnd, albedo), names_to = "band", values_drop_na = TRUE)

ts_custom_long <- ts_custom_wide %>%
    pivot_longer(cols = c(rgnd, albedo), names_to = "band", values_drop_na = TRUE)

ts_glims_long <- ts_glims_wide %>%
    pivot_longer(cols = c(rgnd, albedo), names_to = "band", values_drop_na = TRUE)
```

plot

```
ts_custom_long %>%
  mutate(band = band %>% fct_relevel("rgnd")) %>% # factor RGND before albedo
  ggplot(aes(x = date, y = value, color = platform)) +
  geom_line() +
  facet_grid(rows = vars(band), cols = vars(site), scales = "free")
```



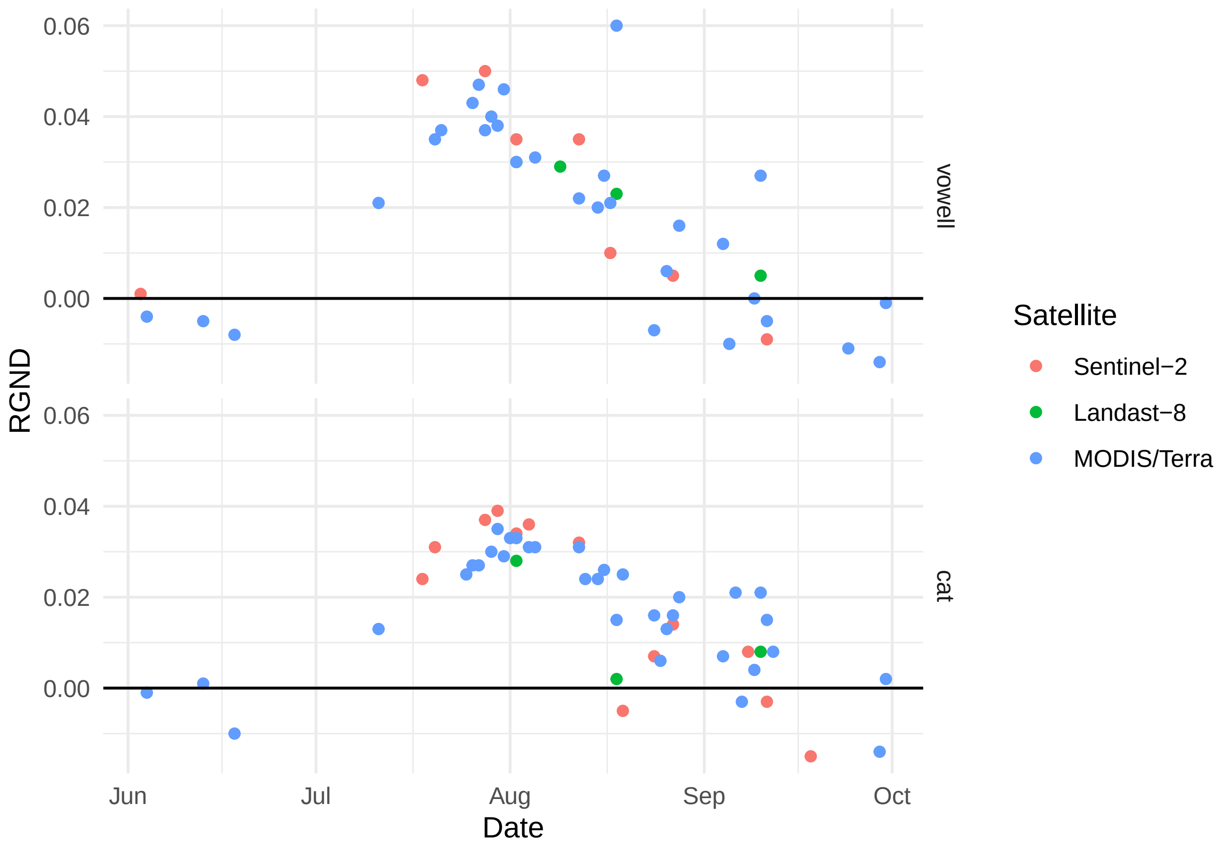
Aqua is very jumpy– Dont show albedo

```r
plot_dat <- ts_custom_wide %>%
  filter(platform %in% c("s2", "l8", "terra")) %>%
  mutate(platform = platform %>%
           fct_relevel(c("s2", "l8", "terra")) %>%# order factors
           fct_recode("Sentinel-2"="s2",
                      "Landast-8"="l8",
                      "MODIS/Terra"="terra"))
plot_dat$platform %>% levels()
```

```
## [1] "Sentinel-2"  "Landast-8"   "MODIS/Terra"
```

```r
pp <- plot_dat %>%
  ggplot(aes(x = date, y = rgnd, color = platform)) +
  facet_grid(rows = vars(site %>% fct_relevel("vowell"))) +
  # MODIS layer
  geom_point(alpha = 0.5, size = 1, data = plot_dat %>%
             filter(platform == "terra") ) +
  # stat_smooth(method = "loess", size = 0,
  #           data = plot_dat %>%
  #             filter(platform == "terra")) +
  # S2 and L8 layers
  # geom_line(alpha = 0.5, size = 1, data = plot_dat %>%
  #             filter(platform %in% c("l8", "s2")) ) +
  geom_point() +
  labs(x = "Date", y = "RGND", color = "Satellite") +
  theme_minimal() +
  geom_hline(yintercept = 0)
pp
```

```
## Warning: Unknown levels in `f`: vowell
```



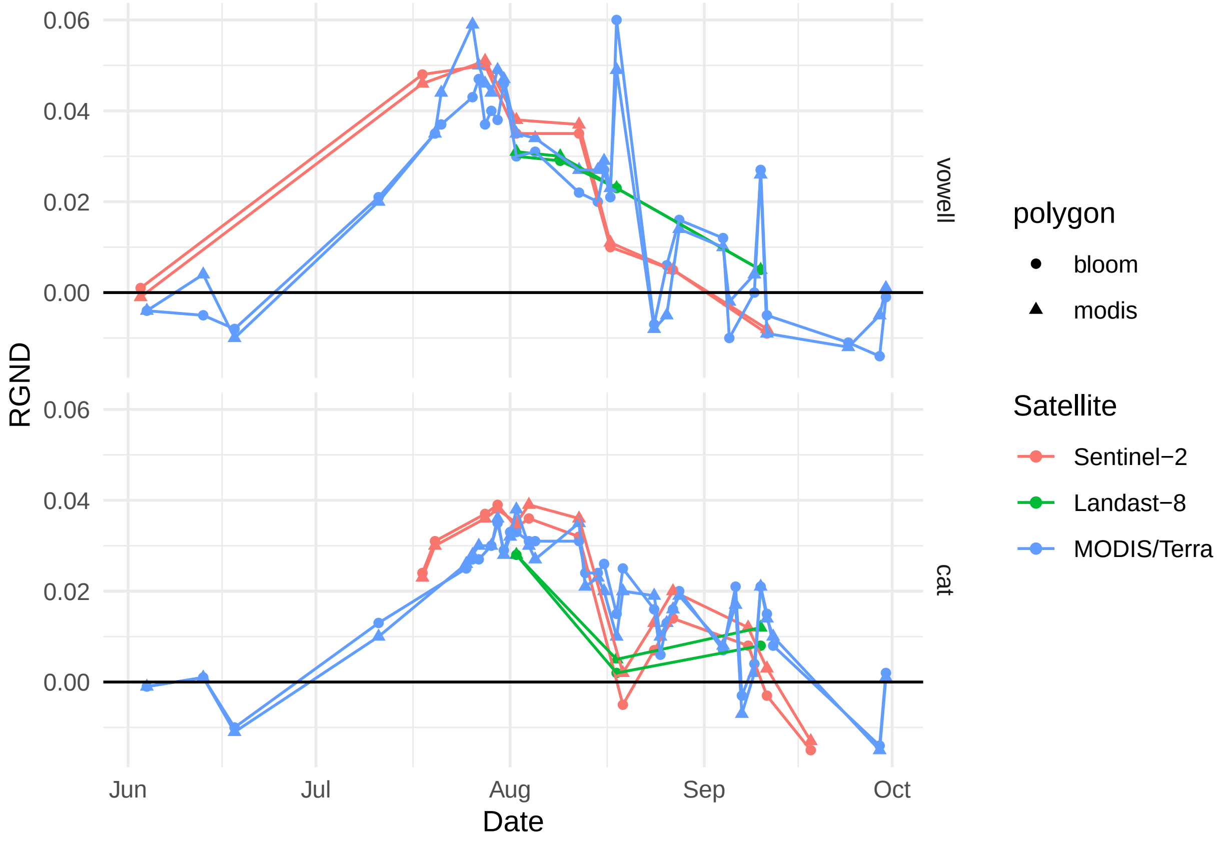
```
# ggsave(here("figs/satellite_time_series.pdf"), height = 5, width = 7)
```

Odd drop in mid August on Cat, not due to clouds. Late season estimates not reliable, bare ice showing.

up to 0.05 on July 30, this corresponds to an estimated cell density of 2.5e8 per 31 days, or a growth rate of +8 million cells per square meter per day.

compare time series polygons

```
plot_dat2 <- ts_custom_wide %>% add_column(polygon = "bloom") %>%
  bind_rows(ts_modis_wide %>% add_column(polygon = "modis")) %>%
  filter(platform %in% c("s2", "l8", "terra")) %>%
  mutate(platform = platform %>%
           fct_relevel(c("s2", "l8", "terra")) %>%# order factors
           fct_recode("Sentinel-2"="s2",
                      "Landast-8"="l8",
                      "MODIS/Terra"="terra"))


pp3 <- plot_dat2 %>%
  ggplot(aes(x = date, y = rgnd, color = platform, shape = polygon)) +
  facet_grid(rows = vars(site %>% fct_relevel("vowell"))) +
  # MODIS layer
  # geom_point(alpha = 0.5, size = 1, data = plot_dat %>%
  #              filter(platform == "terra") ) +
  # stat_smooth(method = "loess", size = 0,
  #           data = plot_dat %>%
  #             filter(platform == "terra")) +
  # S2 and L8 layers
  # geom_line(alpha = 0.5, size = 1, data = plot_dat %>%
  #             filter(platform %in% c("l8", "s2")) ) +
  geom_point() +
  geom_line() +
  labs(x = "Date", y = "RGND", color = "Satellite") +
  theme_minimal() +
  geom_hline(yintercept = 0)
pp3
```



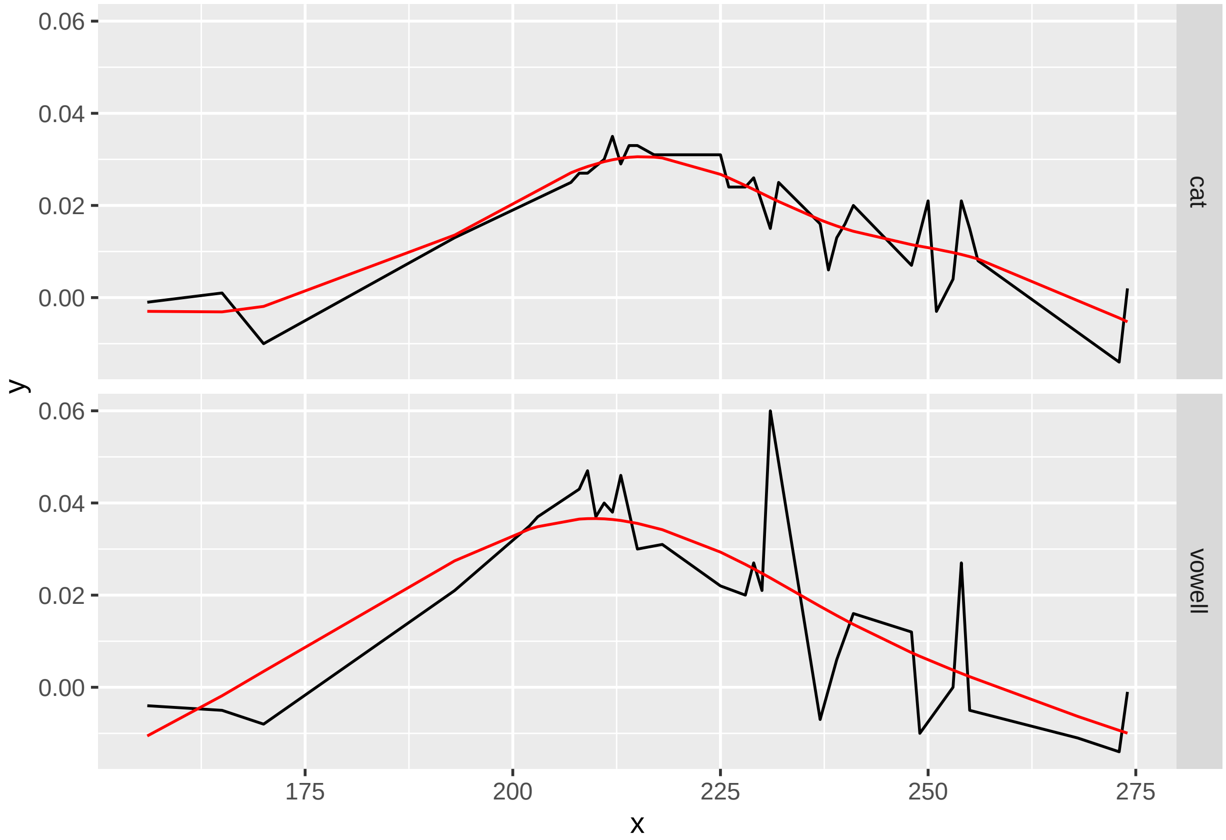
consensus RGND phenology

Approach: establish a "peak bloom" date for each glacier, then take the mean RGND for each platform/site within +/- n days of the peak date.

I will use MODIS to establish peak bloom, since there is more data.

apply smooth spline to Terra

```
nk <- 10

terra <- ts_custom_wide %>%
  filter(platform=="terra") %>%
  group_by(site) %>% # split into vowell and cat time series
  mutate(yday = yday(date)) %>%
  select(platform, site, yday, rgnd)

spl <-
  terra %>%
  nest() %>%
  ungroup() %>% # nest automatically groups.. why?
  mutate(mod = map(data, ~smooth.spline(x=.x$yday, y=.x$rgnd, nknots = nk)),
         aug = map(mod, broom::augment)) %>%
  unnest(cols = c(aug))

# sanity check
spl %>%
  ggplot(aes(x = x)) +
  geom_line(aes(y = y)) +
  geom_line(aes(y = .fitted), color = "red") +
  facet_grid(rows = vars(site))
```



get peak bloom date from terra

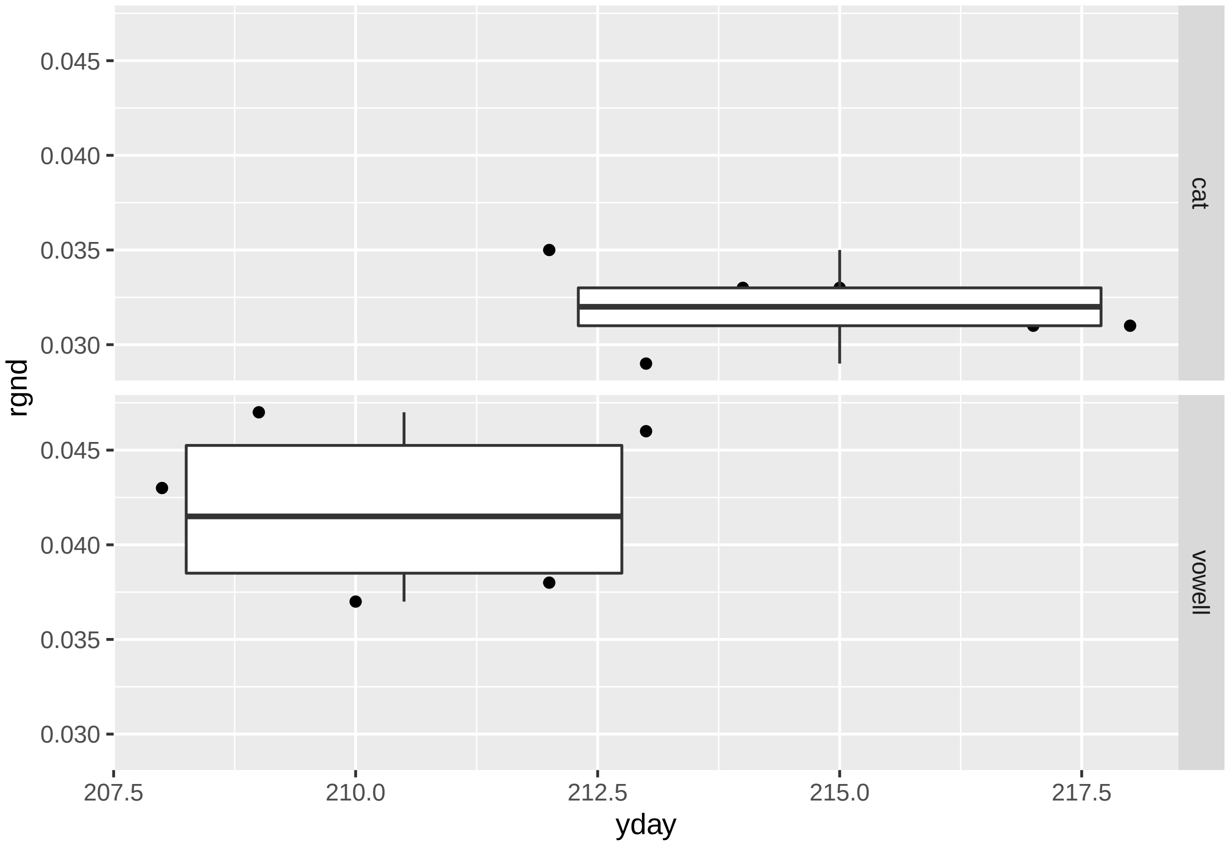
```
max_smooth_yday <- spl %>%
  group_by(site) %>%
  filter(.fitted == max(.fitted)) %>%
  select(site, x)
```

```
window <- 3 # includes

peak_data <- terra %>%
  left_join(max_smooth_yday) %>%
  mutate(min = x - window,
         max = x + window) %>%
  filter(yday >= min,
         yday <= max) %>%
  ungroup() %>%
  select(platform, site, yday, rgnd) # rgnd is the actual data, not smoothed
```

```
## Joining, by = "site"
```

```
peak_data %>%
  ggplot(aes(x = yday, y= rgnd)) +
  geom_point() +
  facet_grid(rows = vars(site)) +
  geom_boxplot()
```

```
## Warning: Continuous x aesthetic -- did you forget aes(group=...)?
```



Peak bloom date: vowell 210 = Jul 29 cat 215 = Aug 3

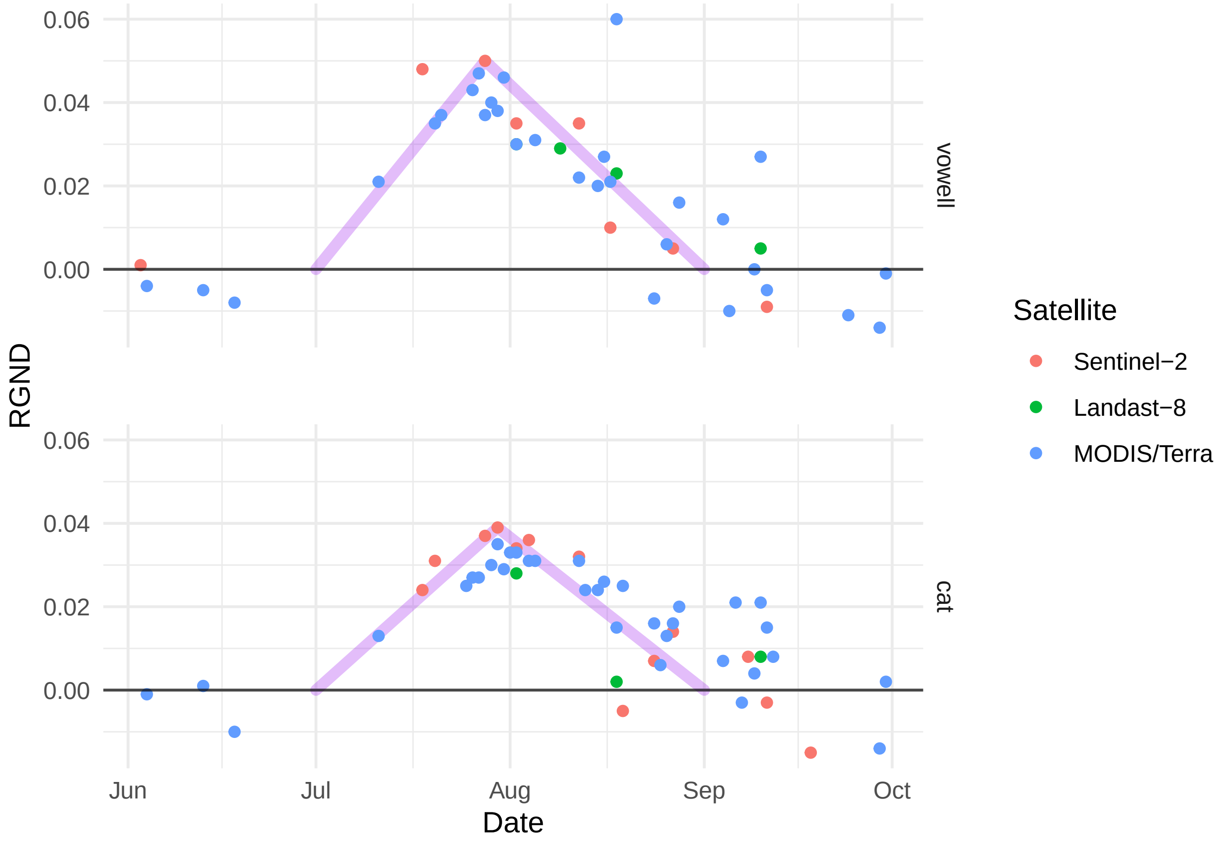
Peak RGND (from S2) vowell: 0.05 (7-28) cat: 0.039 (7-30i)

Peak bloom week: (+/- 3 days of peak bloom day):

vowell 213 208 – jul26 aug1 cat 218 212 – jul31 aug 6

add consensus to plot

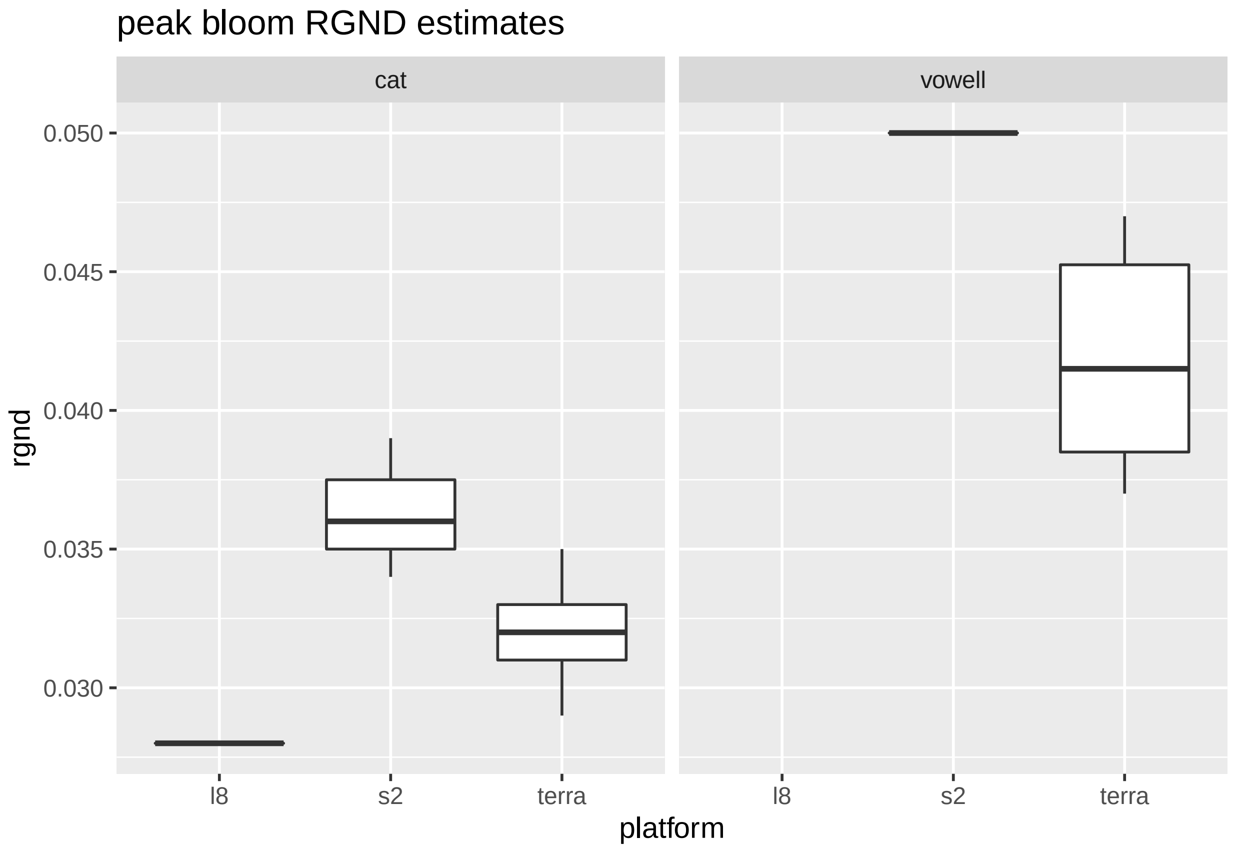
```
cons <- tibble(
  platform = "consensus",
  site = c(rep("vowell", 3), rep("cat", 3)),
  date = ymd(c("2020-07-01", "2020-07-28", "2020-09-01", "2020-07-01", "2020-07-30", "2020-09-01")),
  rgnd = c(0,0.05, 0, 0,0.039, 0)
)


pp2 <- plot_dat %>%
  ggplot(aes(x = date, y = rgnd, color = platform)) +
  facet_grid(rows = vars(site %>% fct_relevel("vowell"))) +
  geom_line(data = cons, color = "purple", size = 2, alpha = 0.3, lineend='round') +
  geom_point() +
  labs(x = "Date", y = "RGND", color = "Satellite") +
  theme_minimal() +
  geom_hline(yintercept = 0, alpha = 0.7) +
  theme(panel.spacing = unit(2, "lines")) #+
  # geom_smooth(method = "lm", formula = y~poly(x,2),
  #             se = FALSE,
  #             data = plot_dat %>%
  #               filter(platform=="Sentinel-2") )

pp2
```



```
# ggsave(here("figs/final_fig5.pdf"), width = 7, height = 5)
```

compare mean RGND estimates between satellites for peak bloom week

```
# get the min/max days from the terra peak data for filtering the rest of the dataset
range <- max_smooth_yday %>%
  mutate(max = x+window, min = x-window)

# filter the L8/S2 data to within this date range
all_peak_data <- ts_custom_wide %>%
  filter(platform %in% c("s2", "l8")) %>%
  mutate(yday = yday(date)) %>%
  left_join(range, by = "site") %>%
  group_by(site, platform) %>%
  filter(yday>=min, yday<=max) %>%
  select(platform, site, yday, rgnd) %>%
  # tack on the terra data
  bind_rows(peak_data) %>%
  ungroup()

all_peak_data %>%
  ggplot(aes(x = platform, y = rgnd)) +
  geom_boxplot() +
  facet_wrap(vars(site)) +
  ggtitle("peak bloom RGND estimates")
```



Sentinel > Terra > Landsat

larger grid cells, signal is dampened by more noise The landsat had some cloud issues, disregard.

In table format, individually

```
means <- all_peak_data %>%
  group_by(platform, site) %>%
  summarise(mean(rgnd), n(), sd(rgnd)) %>%
  ungroup()
```

```
## `summarise()` has grouped output by 'platform'. You can override using the `.groups` argument.
```

```
means
```

```
## # A tibble: 5 x 5
##   platform site   `mean(rgnd)` `n()` `sd(rgnd)`
##   <chr>    <chr>         <dbl> <int>      <dbl>
## 1 l8       cat           0.028     1   NA
## 2 s2       cat           0.0363    3    0.00252
## 3 s2       vowell        0.05      1   NA
## 4 terra    cat           0.032     6    0.00210
## 5 terra    vowell        0.0418    6    0.00417
```

How much lower is Terra from S2?

```
0.05 - 0.041 # 0.009
```

```
## [1] 0.009
```

```
0.036 - 0.032 # 0.004
```

```
## [1] 0.004
```

```
mean(c(0.009, 0.004)) # av difference
```

```
## [1] 0.0065
```

```
0.041/0.05 # .82
```

```
## [1] 0.82
```

```
0.032/0.036 #.888
```

```
## [1] 0.8888889
```

```
mean(c(0.888, .82))
```

```
## [1] 0.854
```

The second of the three S2 images for cat had some cloud issues as well, disregard.

Just the use the max S2 as our best estimate

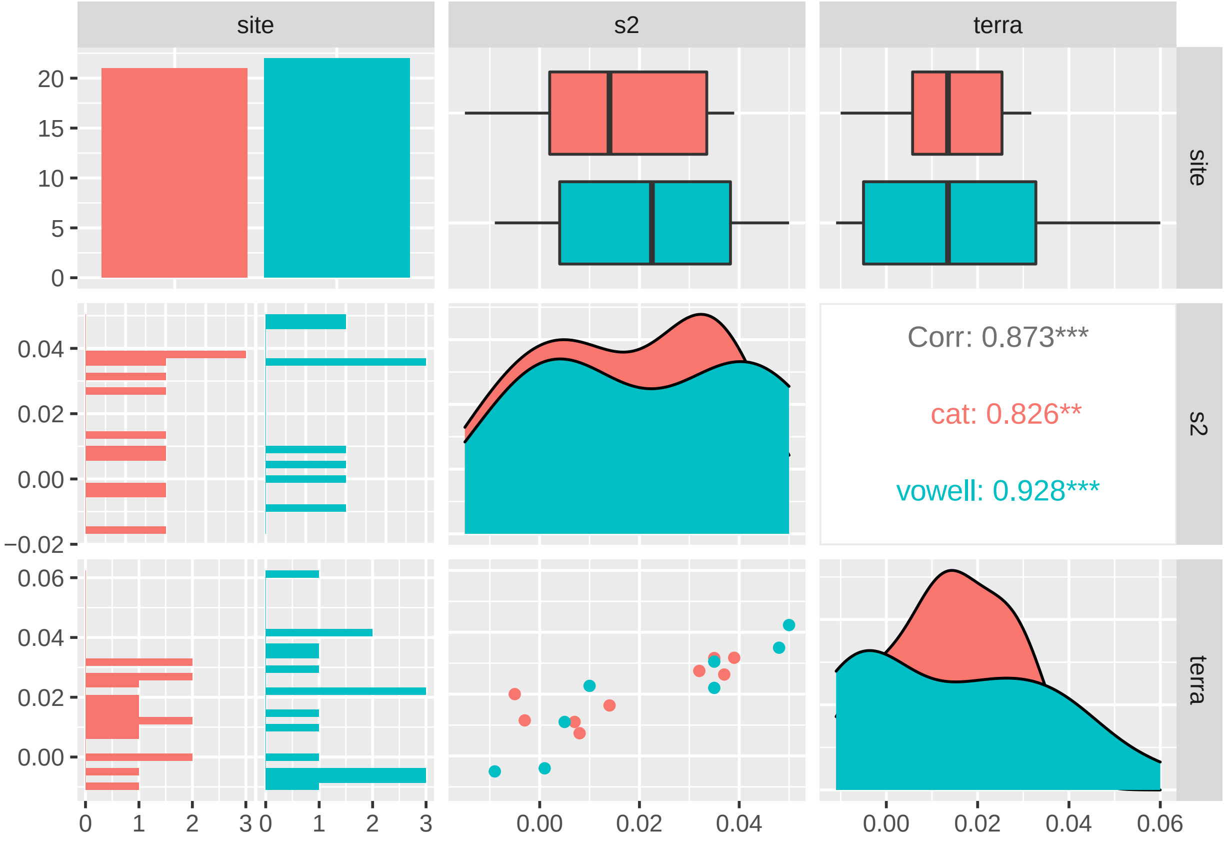
###compare satellite RGND within 1 day window

```
date_bins <- ts_custom_wide %>%
  filter(!platform %>% str_detect("Snow")) %>%
  # bin time into discrete 3 day intervals
  mutate(date_bin = date %>% yday() %>% cut_interval(n=30)) %>%
  group_by(platform, site, date_bin) %>%
  summarise(rgnd = mean(rgnd)) %>%
  ungroup() %>%
  pivot_wider(names_from = "platform", values_from = "rgnd")
```

```
## `summarise()` has grouped output by 'platform', 'site'. You can override using the `.groups` argumen
```

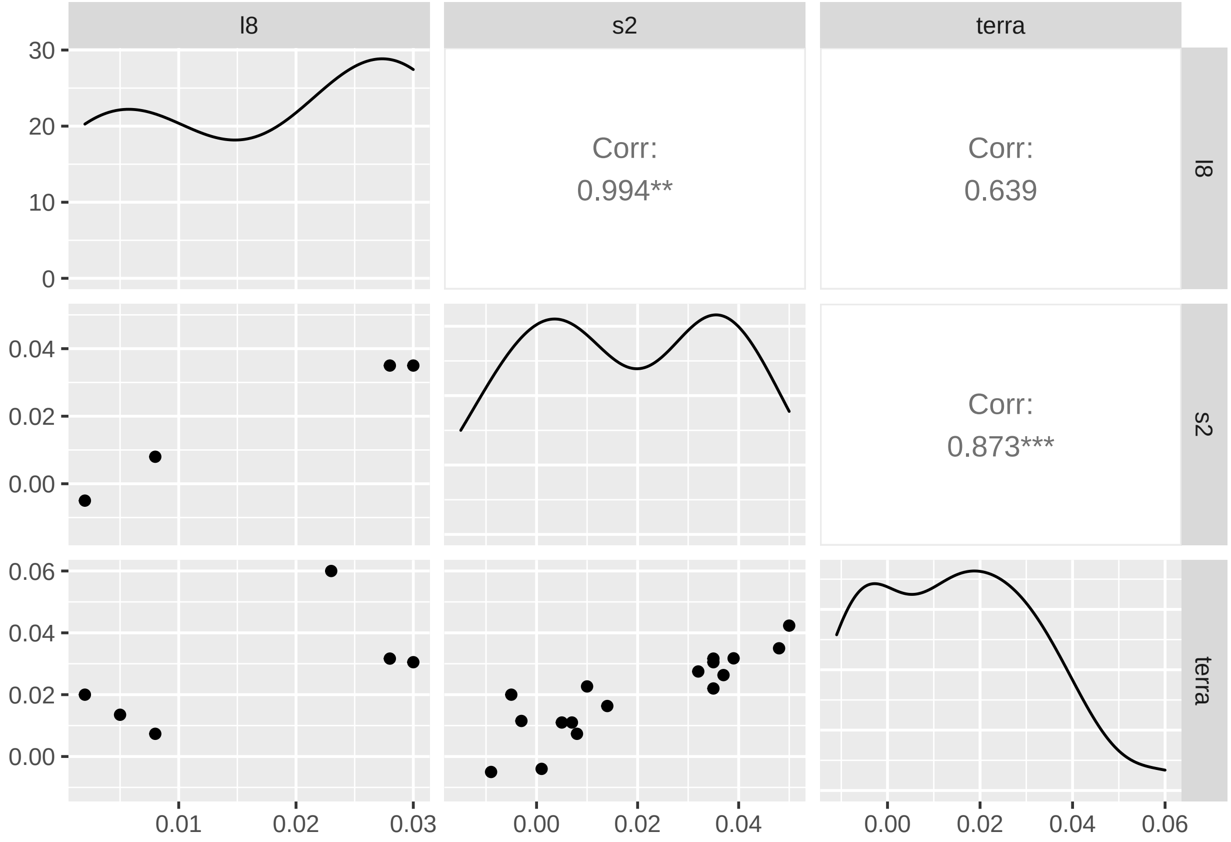
```
date_bins %>%
  select(-date_bin, -aqua, -18) %>%
  ggpairs(colums = 2:4, ggplot2::aes(colour=site))
```

```
## Warning in warn_if_args_exist(list(...)): Extra arguments: 'colums' are being
## ignored. If these are meant to be aesthetics, submit them using the 'mapping'
## variable within ggpairs with ggplot2::aes or ggplot2::aes_string.

## Warning: Removed 24 rows containing non-finite values (stat_boxplot).

## Warning: Removed 8 rows containing non-finite values (stat_boxplot).

## `stat_bin()` using `bins = 30`. Pick better value with `binwidth`.

## Warning: Removed 24 rows containing non-finite values (stat_bin).

## Warning: Removed 24 rows containing non-finite values (stat_density).

## Warning in ggally_statistic(data = data, mapping = mapping, na.rm = na.rm, :
## Removed 26 rows containing missing values

## `stat_bin()` using `bins = 30`. Pick better value with `binwidth`.

## Warning: Removed 8 rows containing non-finite values (stat_bin).

## Warning: Removed 26 rows containing missing values (geom_point).

## Warning: Removed 8 rows containing non-finite values (stat_density).
```



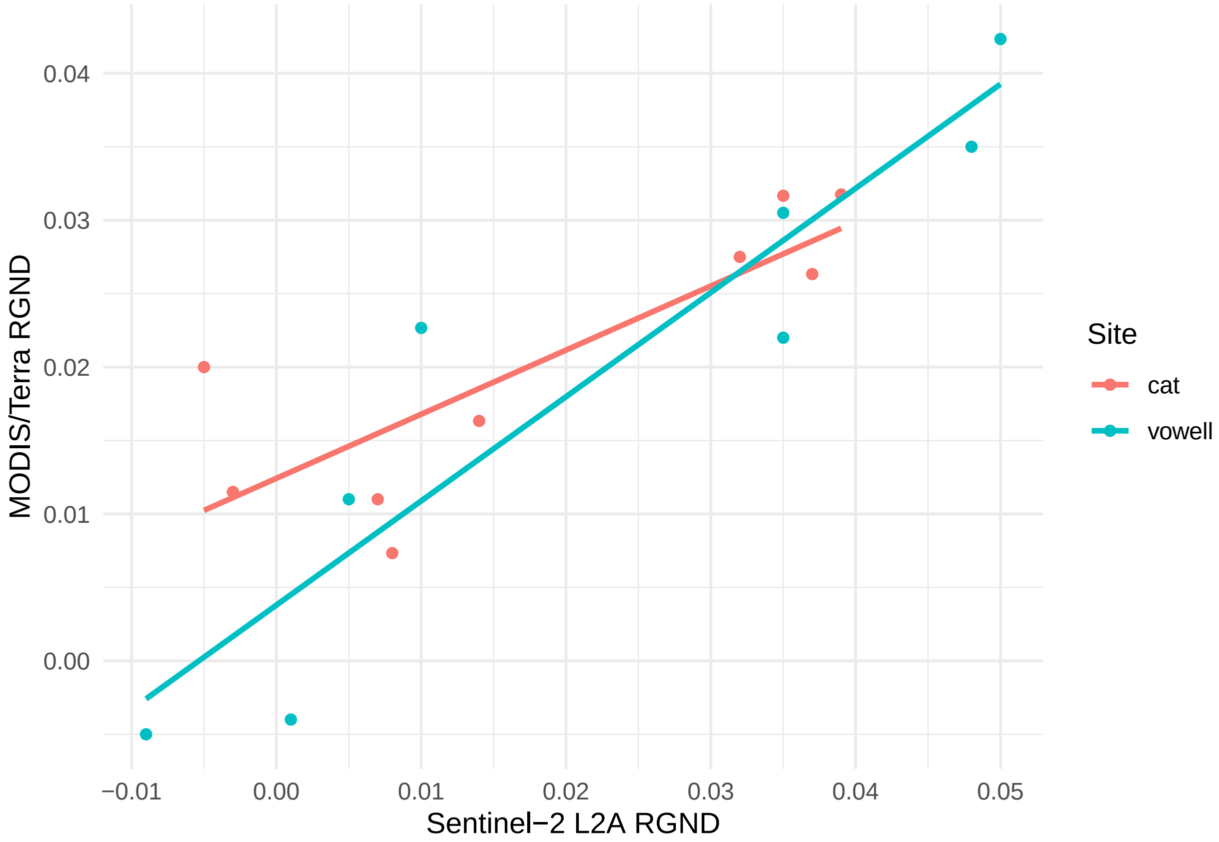
```
date_bins %>%
  select(-date_bin, -site, -aqua) %>%
  ggpairs(colums = 2:4)
```

```
## Warning in warn_if_args_exist(list(...)): Extra arguments: 'colums' are being
## ignored. If these are meant to be aesthetics, submit them using the 'mapping'
## variable within ggpairs with ggplot2::aes or ggplot2::aes_string.

## Warning: Removed 36 rows containing non-finite values (stat_density).

## Warning in ggally_statistic(data = data, mapping = mapping, na.rm = na.rm, :
## Removed 39 rows containing missing values

## Warning in ggally_statistic(data = data, mapping = mapping, na.rm = na.rm, :
## Removed 37 rows containing missing values

## Warning: Removed 39 rows containing missing values (geom_point).

## Warning: Removed 24 rows containing non-finite values (stat_density).

## Warning in ggally_statistic(data = data, mapping = mapping, na.rm = na.rm, :
## Removed 26 rows containing missing values

## Warning: Removed 37 rows containing missing values (geom_point).

## Warning: Removed 26 rows containing missing values (geom_point).

## Warning: Removed 8 rows containing non-finite values (stat_density).
```



```
date_bins %>%
  select(site, s2, terra) %>%
  drop_na() %>%
  ggplot(aes(x = s2, y = terra, color = site)) +
  geom_point() +
  geom_smooth(method = "lm", se=FALSE) +
  labs(x = "Sentinel-2 L2A RGND", y="MODIS/Terra RGND", color = "Site") +
  theme_minimal()
```

```
## `geom_smooth()` using formula 'y ~ x'
```



Glacier stats

whole glacier

```
# approach: take mean RGND over glacier area, apply algae abundance models, multiply by glacier area
# this should give the same result as individually summing each grid cell's algae productivity

max_algae_glims <-
  ts_glims_wide %>%
  filter(platform == "s2") %>%
  group_by(site) %>%
  summarise(max_rgnd = max(rgnd, na.rm = TRUE)) %>%
  # applying the algae abundance models
  mutate(max_frac_red = max_rgnd * 154, # percent cell area
         max_cell_density = max_rgnd * 5.4, # bil cells / m2
         max_toc = 12.9*max_rgnd, # g/m2
```

name	vowell	cat
mean_rgnd	2.20e-02	2.30e-02
cell_area_percent_cov	3.39e+00	3.54e+00
cell_density_bil_cell_per_m2	1.19e-01	1.24e-01
toc_g_m2	2.84e-01	2.97e-01
tn_mg_m2	8.27e+00	8.65e+00
delta_albedo	1.94e-02	2.02e-02
polygon_area_m2	6.04e+06	5.85e+06
total_cells	7.18e+14	7.27e+14
total_toc_kg	1.71e+03	1.74e+03
total_tn_kg	5.00e+01	5.06e+01

```
         max_tn = 376*max_rgnd, # mg/m2
         max_delta_albedo = 0.88 * max_rgnd, # on the whole glacier
         glacier_area_m2 = c(5.85e6, 6.04e6), # m2
         total_cells = max_cell_density * glacier_area_m2 * 1e9,
         total_toc_kg = (max_toc * glacier_area_m2)/1000, # convert from g to kg
         total_tn_kg = (max_tn * glacier_area_m2)/(1000*1000)) # convert from mg to kg

tt <- max_algae_glims %>%
  arrange(max_delta_albedo) %>%
  rename(mean_rgnd = max_rgnd, cell_area_percent_cov = max_frac_red,
         cell_density_bil_cell_per_m2 = max_cell_density,
         toc_g_m2 = max_toc,
         tn_mg_m2 = max_tn,
         delta_albedo = max_delta_albedo,
         polygon_area_m2 = glacier_area_m2) %>%
  pivot_longer(-site) %>%
  pivot_wider(names_from = site) %>%
  # control number of sig figs
  mutate_if(is.numeric, format, digits=3,nsmall = 0)
tt %>%
  kbl(format = "latex", booktabs = TRUE) %>%
  kable_styling(latex_options = "striped")
```

At peak bloom: Vowell: 2.5% coverage, albedo -1.4%, 1245 kg TOC and 36 kg TN Catamount: 3.2% coverage, albedo -1.8%, 1585 kg TOC, 46 kg TN

Applying G-D melt model, algae melt = 0.13 sqrt(area in cm2/l)

```
melt <- function(rgnd){
  cell_dens <- rgnd * 1234 # my model, in cm2/L
  0.13*sqrt(cell_dens)
}
melt(0.016) # vowell
```

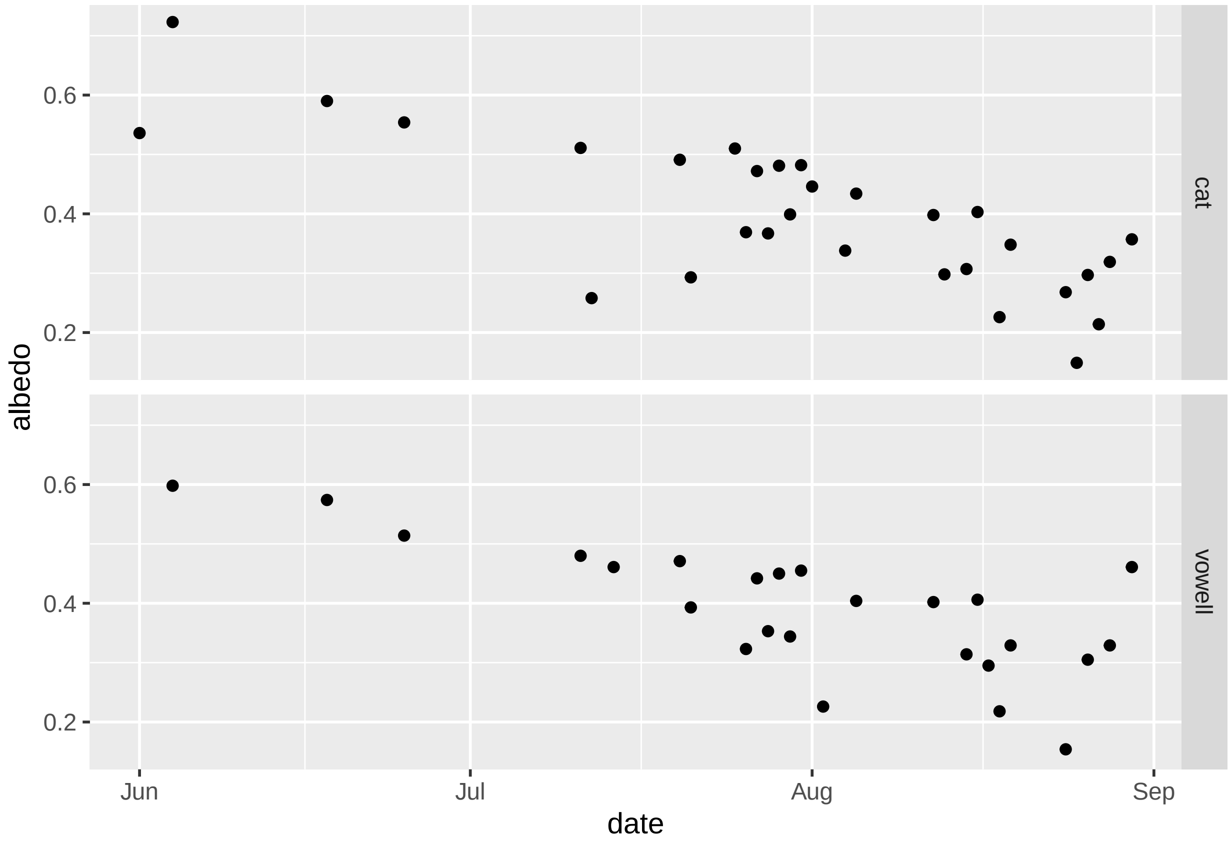
```
## [1] 0.5776449
```

```
melt(0.021) # cat
```

```
## [1] 0.6617753
```

Is the estimated delta albedo reasonable? check the MODIS albedo data

```
ts_glims_wide %>%
  filter(date<"2020-09-01") %>%
  drop_na(albedo) %>%
  ggplot(aes(x = date, y = albedo)) +
  geom_point() +
  facet_grid(rows = vars(site)) #+
```



```
  # geom_smooth(method = "lm", se = FALSE)
```

The first point in July is ~0.5. The last points in late July oscillate around 0.4. so let's say MODIS estimates a delta albedo of 0.5 - 0.4 = 0.1 (July), or 10%

This is much higher than our estimate of 1-2%, but we would expect some decline in albedo in the absence of snow algae. so 8-9% albedo decline is due to snowmelt?

radiative forcing

Approximate bloom density as abs function to integrate RF over the summer (Jul - Aug)

```
my_site <- "cat"

max_delta_albedo <- max_algae_glims %>%
  filter(site == my_site) %>%
  pull(max_delta_albedo)




# delta_albedo_approx %>% ggplot(aes(x = date, y= delta_albedo)) + geom_line()
# srad_clean %>% ggplot(aes(x = date, y = srad)) + geom_line() +facet_grid(rows = vars(site))

# srad_clean %>%
#   filter(site==my_site) %>%
#   select(date, srad) %>%
#   left_join(delta_albedo_approx, by = "date") %>%
#   mutate(rf = srad * delta_albedo) %>%
#   summarise(rf = mean(rf))
```

Vowell bloom has a RF of 2.98 W/m2 from July - August averaged over the entire glacier area Cat was 3.85

just the "bloom area"

```
max_algae_custom <-
  ts_custom_wide %>%
  filter(platform == "s2") %>%
  group_by(site) %>%
  summarise(max_rgnd = max(rgnd, na.rm = TRUE)) %>%
  # applying the algae abundance models
  mutate(max_frac_red = max_rgnd * 154, # percent cell area
         max_cell_density = max_rgnd * 5.4, # bil cells / m2
         max_toc = 12.9*max_rgnd, # g/m2
         max_tn = 376*max_rgnd, # mg/m2
         max_delta_albedo = 0.88 * max_rgnd,
         bloom_area_m2 = c(1.9e6, 2e6), # m2
         total_cells = max_cell_density * bloom_area_m2 * 1e9,
         total_toc_kg = (max_toc * bloom_area_m2)/1000, # convert from g to kg
         total_tn_kg = (max_tn * bloom_area_m2)/(1000*1000)) # convert from mg to kg

tt2 <- max_algae_custom %>%
  arrange(max_delta_albedo) %>%
  rename(mean_rgnd = max_rgnd,
         cell_area_percent_cov = max_frac_red,
         cell_density_bil_cell_per_m2 = max_cell_density,
         toc_g_m2 = max_toc,
         tn_mg_m2 = max_tn,
         delta_albedo = max_delta_albedo,
         polygon_area_m2 = bloom_area_m2) %>%
  pivot_longer(-site) %>%
  pivot_wider(names_from = site) %>%
  select(name, vowell, cat) %>%
  # control number of sig figs
  mutate_if(is.numeric, format, digits=3,nsmall = 0)
tt2 %>%
  kbl(format = "latex", booktabs = TRUE) %>%
  kable_styling(latex_options = "striped")
```

delta albedo is the amount by which our RGND-albedo model predicts snow algae to decrease albedo (Fig 3B)

total abundance: bloom polygon likely underestimate (boundaries too small) glacier polygon likely overestimate of total prod (false positives)

name	vowell	cat
mean_rgnd	5.00e-02	3.90e-02
cell_area_percent_cov	7.70e+00	6.01e+00
cell_density_bil_cell_per_m2	2.70e-01	2.11e-01
toc_g_m2	6.45e-01	5.03e-01
tn_mg_m2	1.88e+01	1.47e+01
delta_albedo	4.40e-02	3.43e-02
polygon_area_m2	2.00e+06	1.90e+06
total_cells	5.40e+14	4.00e+14
total_toc_kg	1.29e+03	9.56e+02
total_tn_kg	3.76e+01	2.79e+01

```
melt(0.05)# 10.2 mm SWE/day
```

```
## [1] 1.021142
```

```
melt(0.039) #9 mm SWE/day
```

```
## [1] 0.9018478
```

1 cm water over the entire bloom polygon area, is a volume of

```
# in units of m2 and m
1.6e6*0.012 # 19,000 m3/day
```

```
## [1] 19200
```

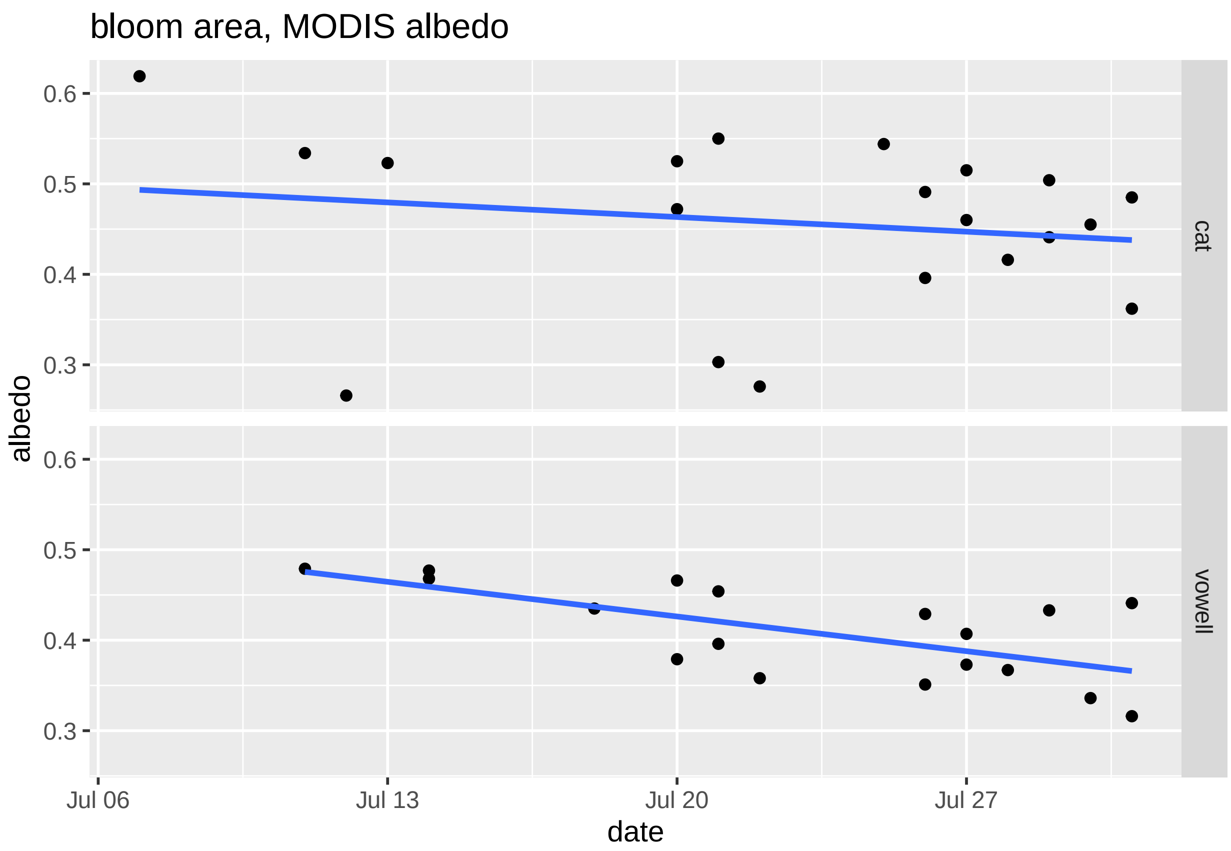
```
2.1e6*0.009 # 19,000 m3 per day
```

```
## [1] 18900
```

Compare to the MODIS albedo (using the custom polygon)

```
ts_custom_wide %>%
  filter(date<"2020-08-01", date>"2020-07-01") %>%
  drop_na(albedo) %>%
  ggplot(aes(x = date, y = albedo)) +
  geom_point() +
  facet_grid(rows = vars(site)) +
  ggtitle("bloom area, MODIS albedo") +
  geom_smooth(method="lm", se=FALSE)
```

```
## `geom_smooth()` using formula 'y ~ x'
```



also looks like a 10% drop in albedo in July. But this is very few grid cells, at this resolution hard to say. If our model predicts a drop of ~4%. snowdepth could effect this. showing dark ice beneath. What was the depth under our albedo measurements? but three sites should average this out…

fill in missing vals from consensus

expand

```
cons2 <- cons %>% # from consensus for bloom polygon
  complete(site, date = seq(min(date), max(date), by = "day")) %>%
  group_by(site) %>%
  mutate(platform = "consensus",
         rgnd = zoo::na.approx(rgnd),
         delta_albedo = 0.88 * rgnd)
cons2
```

```
## # A tibble: 126 x 5
## # Groups:   site [2]
##    site  date       platform     rgnd delta_albedo
##    <chr> <date>     <chr>       <dbl>        <dbl>
##  1 cat   2020-07-01 consensus 0             0
##  2 cat   2020-07-02 consensus 0.00134       0.00118
##  3 cat   2020-07-03 consensus 0.00269       0.00237
##  4 cat   2020-07-04 consensus 0.00403       0.00355
##  5 cat   2020-07-05 consensus 0.00538       0.00473
##  6 cat   2020-07-06 consensus 0.00672       0.00592
##  7 cat   2020-07-07 consensus 0.00807       0.00710
##  8 cat   2020-07-08 consensus 0.00941       0.00828
##  9 cat   2020-07-09 consensus 0.0108       0.00947
## 10 cat   2020-07-10 consensus 0.0121       0.0107
## # ... with 116 more rows
```

Radiative forcing estimate for bloom polygon

```
srad_clean %>%
  select(date, srad) %>%
  left_join(cons2, by = "date") %>%
  mutate(rf = srad * delta_albedo) %>%
  group_by(site) %>%
  summarise(rf = mean(rf))
```

```
## # A tibble: 3 x 2
##   site      rf
##   <chr>  <dbl>
## 1 cat     7.14
## 2 vowell  9.21
## 3 <NA>   NA
```

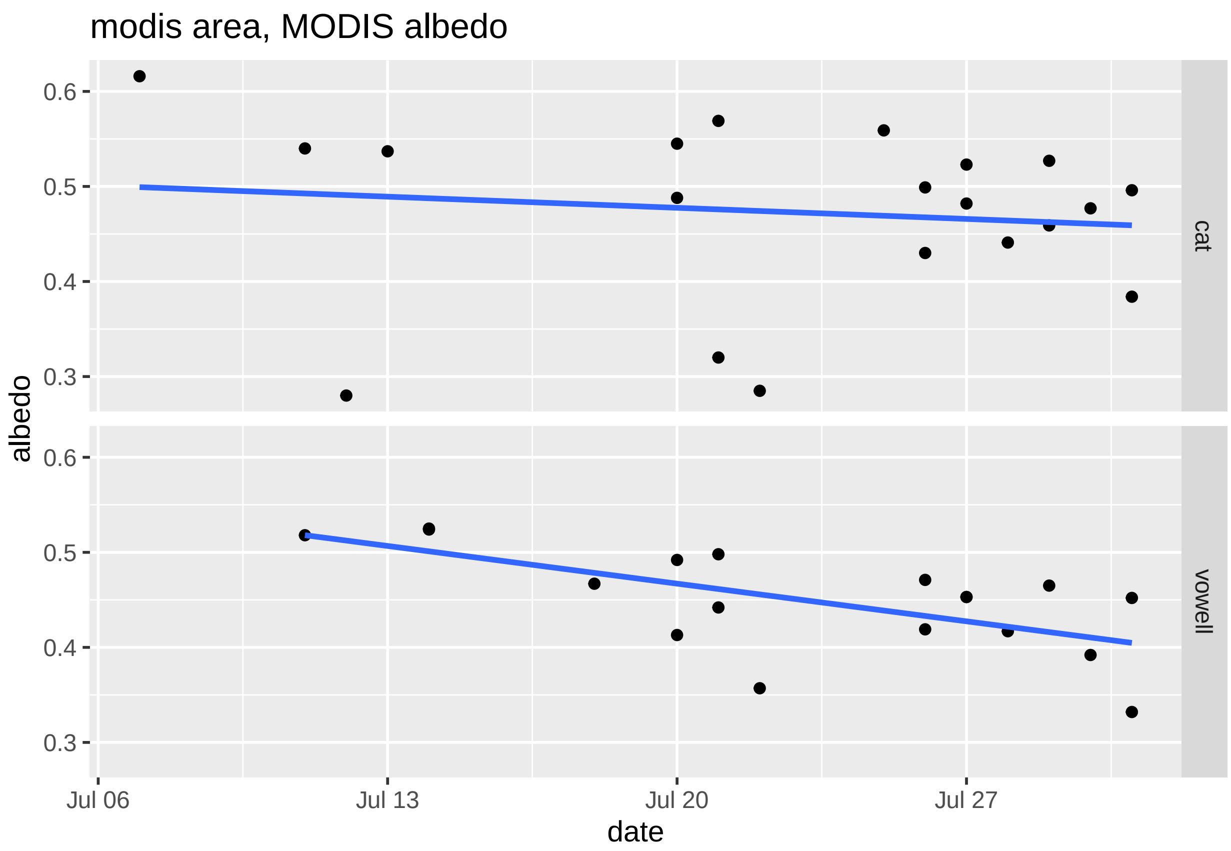
Just over the bloom area, RF of 9.3 and 7.2 over vowell and catamount blooms respectively from Jul-Aug

MODIS polygons

Compare the MODIS albedo vs the predicted algae albedo effect (delta albedo) Might be less of a discrepancy omitting the MODIS grid cells that overlap with seasonal snow cover off the glacier

```
ts_modis_wide %>%
  filter(date<"2020-08-01", date>"2020-07-01") %>%
  drop_na(albedo) %>%
  ggplot(aes(x = date, y = albedo)) +
  geom_point() +
  facet_grid(rows = vars(site)) +
  ggtitle("modis area, MODIS albedo") +
  geom_smooth(method="lm", se=FALSE)
```

```
## `geom_smooth()` using formula 'y ~ x'
```



This looks pretty much the same

```
max_algae_modis <- ts_modis_wide %>%
  filter(platform %>% str_detect( "s2")) %>% # forgot to clip off the ".csv"
  group_by(site) %>%
  summarise(max_rgnd = max(rgnd, na.rm = TRUE)) %>%
  # applying the algae abundance models
  mutate(max_frac_red = max_rgnd * 154, # percent cell area
         max_cell_density = max_rgnd * 5.4, # bil cells / m2
         max_toc = 12.9*max_rgnd, # g/m2
         max_tn = 376*max_rgnd, # mg/m2
         max_delta_albedo = 0.88 * max_rgnd,
         bloom_area_m2 = c(1.9e6, 2e6), # m2
         total_cells = max_cell_density * bloom_area_m2 * 1e9,
         total_toc_kg = (max_toc * bloom_area_m2)/1000, # convert from g to kg
         total_tn_kg = (max_tn * bloom_area_m2)/(1000*1000)) # convert from mg to kg

max_algae_modis %>% arrange(max_delta_albedo)
```

```
## # A tibble: 2 x 11
##   site   max_rgnd max_frac_red max_cell_density max_toc max_tn max_delta_albedo
##   <chr>     <dbl>        <dbl>            <dbl>   <dbl>  <dbl>            <dbl>
## 1 cat       0.039         6.01            0.211   0.503   14.7           0.0343
## 2 vowell    0.051         7.85            0.275   0.658   19.2           0.0449
## # ... with 4 more variables: bloom_area_m2 <dbl>, total_cells <dbl>,
## #   total_toc_kg <dbl>, total_tn_kg <dbl>
```

Same. . .

compare custom polygon with MODIS grid cells

```
# compare <- ts2_wide %>%
#   rename(rgnd_custom_poly = rgnd, albedo_custom_poly = albedo) %>%
#   left_join(ts_wide)
#
# compare %>%
#   ggplot(aes(x = rgnd, y = rgnd_custom_poly)) +
#   geom_point() +
#   facet_wrap(vars(platform, site), scales = "free")
```

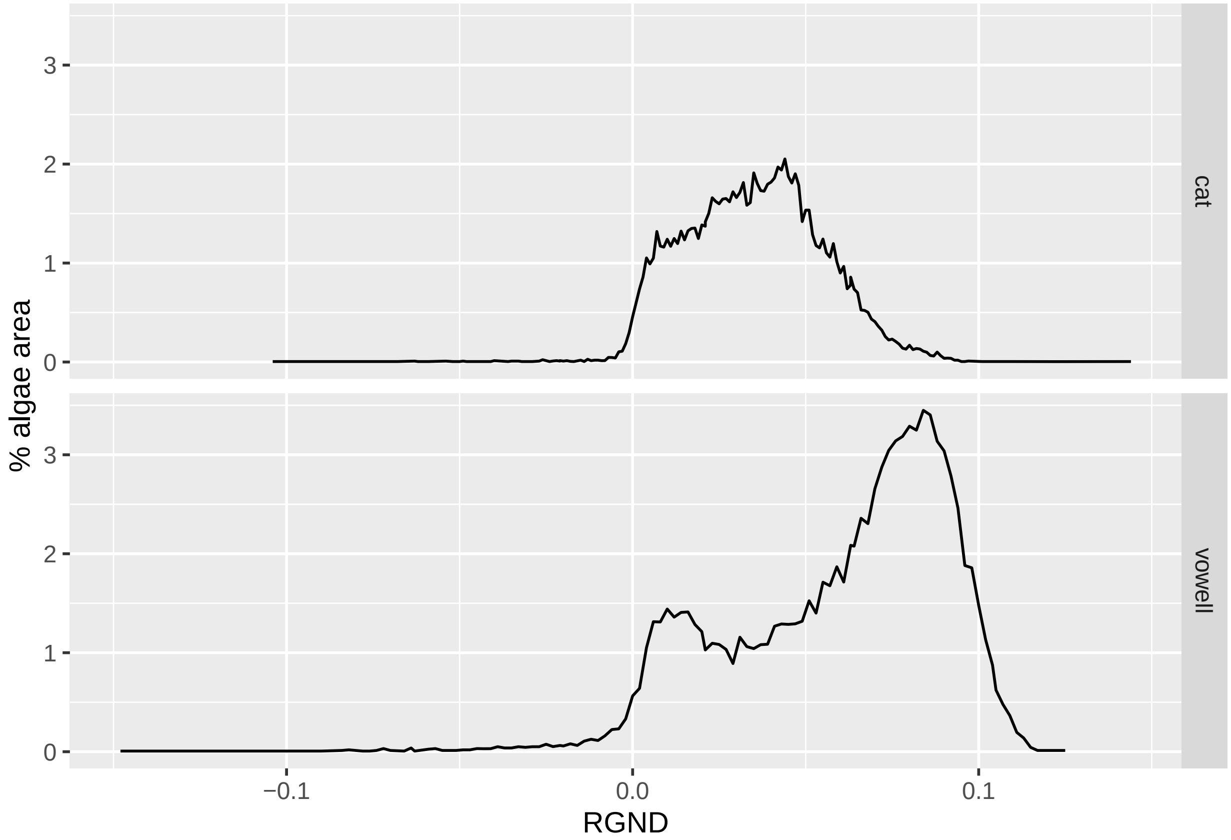
RGND histogram (s2 density plot)

```
grid_cell_count <- hist_clean %>%
  group_by(site) %>%
  summarise(grid_cell_count = sum(rgnd_count, na.rm = TRUE))
grid_cell_count
```

```
## # A tibble: 2 x 2
##   site   grid_cell_count
##   <chr>            <dbl>
## 1 cat             21547.
## 2 vowell          15894.
```

```
hist_clean %>%
  drop_na(rgnd_count) %>% # remove empty bins
  left_join(grid_cell_count) %>%
  mutate(percent_algae_area = (rgnd_count / grid_cell_count) *100) %>%
  ggplot(aes(x = band_value, y = percent_algae_area)) +
  geom_line() +
  facet_grid(rows = vars(site)) +
  labs(y = "% algae area", x = "RGND")
```

```
## Joining, by = "site"
```



```
# reverse engineer the raw obs, hacky
raw_vals_vec_cat <- hist_clean %>%
  filter(site == "cat") %>%
  drop_na(rgnd_count) %>%
  rowwise() %>%
  mutate(reps = list(rep(band_value, rgnd_count))) %>% # ->x;x[9,4] %>% pull() # sanity check
  select(reps) %>%
  unlist() # as list

raw_vals_vec_vowell <- hist_clean %>%
  filter(site == "vowell") %>%
  drop_na(rgnd_count) %>%
  rowwise() %>%
  mutate(reps = list(rep(band_value, rgnd_count))) %>% # ->x;x[9,4] %>% pull() # sanity check
  select(reps) %>%
  unlist() # as list

raw_vals_tbl <- tibble(rgnd = c(raw_vals_vec_cat, raw_vals_vec_vowell),
                       site = c(rep("cat", length(raw_vals_vec_cat)),
                                rep("vowell", length(raw_vals_vec_vowell))) )

raw_vals_tbl %>%
  group_by(site) %>%
  summarise(mean = mean(rgnd), max = max(rgnd), min = min(rgnd), sd = sd(rgnd)) %>%
  mutate(mean_cell_percent = mean * 154,
         mean_toc = 12.9*mean,
         mean_cell = mean * 5.4*mean)
```

```
## # A tibble: 2 x 8
##   site      mean   max    min     sd mean_cell_percent mean_toc mean_cell
##   <chr>    <dbl> <dbl>  <dbl>  <dbl>             <dbl>    <dbl>     <dbl>
## 1 cat     0.0350 0.144 -0.104 0.0197              5.38    0.451   0.00660
## 2 vowell  0.0601 0.125 -0.148 0.0313              9.26    0.776   0.0195
```

```
# # santiy check: compute mean manually from the histogram data
# hist_clean %>%
#   drop_na(rgnd_count) %>%
#   left_join(grid_cell_count) %>%
#   group_by(site) %>%
#   mutate(weighted = band_value*rgnd_count) %>%
#   summarise(mean = sum(weighted)/grid_cell_count) %>%
#   distinct() # cat is 0.034, vowell is 0.06
```

~99% (2 SD) of values from 0 to 0.12 (vowell), and -0.005 to 0.075 0.06-0.03*2 0.035-0.02

During our Catamount transect we observed small patches of concentrated red snow ~10-100 cm wide interspersed on white snow (Supp), but this patchiness was less apparent in Sentinel-2 images due to 10 m spatial resolution

weather/albedo metadata

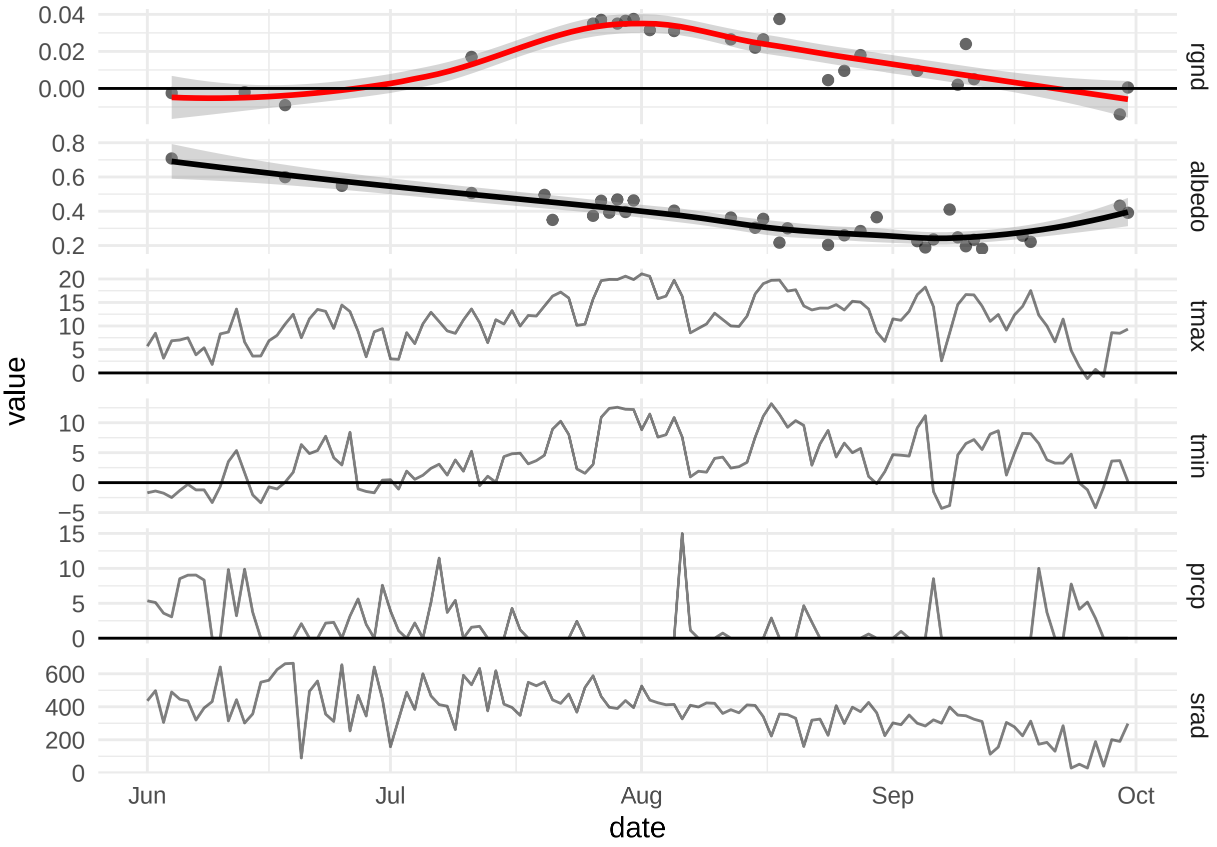
```
my_wx <- ts_custom_wide %>%
  filter(platform %in% c("terra", "terraSnow")) %>%
  select(site, date, rgnd, albedo) %>%
  pivot_longer(-site:-date, values_drop_na =TRUE) %>%
  pivot_wider() %>%
  full_join(wx_clean) %>%
  # mean of both glaciers
  group_by(date) %>%
  select(-site) %>%
  summarise(across(everything(), list(mean = mean), .names = "{.col}")) %>%
  pivot_longer(-date) %>%
  mutate(name = name %>% fct_relevel("rgnd", "albedo", "tmax", "tmin", "prcp", "srad"))
```

```
## Joining, by = c("site", "date")
```

```
my_wx %>%
  ggplot(aes(x = date, y = value)) +
  facet_grid(rows = vars(name), scales = "free") +
  geom_point(alpha = 0.6, data = my_wx %>% filter(name %in% c("rgnd", "albedo"))) +
  theme_minimal() +
  geom_smooth(color = "red",
              data = my_wx %>%
                filter(name =="rgnd") %>%
                # set Jul 1 to 0
                bind_rows(tibble(date = ymd("2020-07-01", "2020-07-02", "2020-07-03", "2020-07-04", "202
                                 name = "rgnd", value=0)) %>%
                mutate(name = name %>% fct_relevel("rgnd"))) +
  geom_smooth(data = my_wx %>% filter(name =="albedo"), color = "black") +
  geom_line(alpha = 0.5, data = my_wx %>% filter(!(name %in% c("rgnd", "albedo")))) +
```

```r
  # add horiz line to some facets
  geom_hline(aes(yintercept = 0), data = my_wx %>% filter(!(name %in% c("albedo", "srad"))) )
```

```
## `geom_smooth()` using method = 'loess' and formula 'y ~ x'

## Warning: Removed 98 rows containing non-finite values (stat_smooth).

## `geom_smooth()` using method = 'loess' and formula 'y ~ x'

## Warning: Removed 88 rows containing non-finite values (stat_smooth).

## Warning: Removed 186 rows containing missing values (geom_point).
```


rgnd
albedo
tmax
tmin
prcp
srad
value
Jun
Jul
Aug
Sep
Oct
date