library(tidyverse)

download.file("<ftp://sidads.colorado.edu/DATASETS/NOAA/G02135/north/daily/data/N_seaice_extent_daily_v3.0.csv>",

"Downloads/arctic\_ice.csv")

df\_names = read\_csv("Downloads/arctic\_ice.csv",

col\_names = F,

n\_max = 1)

df = read\_csv("Downloads/arctic\_ice.csv",

skip = 2,

col\_names = as.character(df\_names)) %>%

janitor::clean\_names() %>%

mutate(date = paste(year, month, day, sep = "-"),

date = as.Date(date),

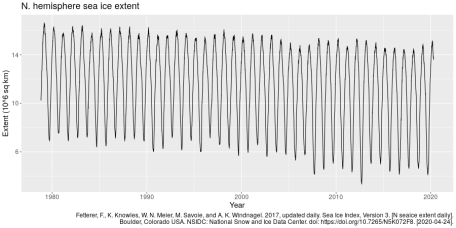
extent = replace(extent, missing > 0, NA)) %>%

select(date, extent)

wrap\_width = scales::wrap\_format(150)

ice\_cite = wrap\_width("Fetterer, F., K. Knowles, W. N. Meier, M. Savoie, and A. K. Windnagel. 2017, updated daily. Sea Ice Index, Version 3. [N seaice extent daily]. Boulder, Colorado USA. NSIDC: National Snow and Ice Data Center. doi: <https://doi.org/10.7265/N5K072F8>. [2020-04-24].")

We’ve now got a two column tibble containing date and sea ice extent. We can see this by plotting our data. (Something similar can be achieved with base graphics using plot(df)):



ggplot(df, aes(date, extent)) +

geom\_line() +

labs(title = "N. hemisphere sea ice extent",

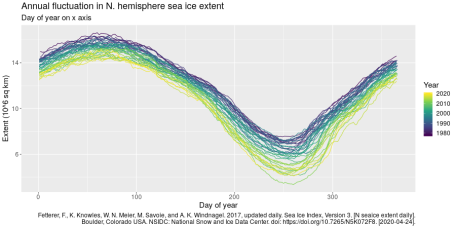
x = "Year",

y = "Extent (10^6 sq km)",

caption = ice\_cite) +

theme(text = element\_text(size = 15))

What do our data look like when we overlay years? i.e. the problem posed at the beginning of this post.



df %>%

mutate(year = year(date),

date = yday(date)) %>%

ggplot(aes(date, extent,

group = year,

colour = year)) +

geom\_line() +

scale\_colour\_viridis\_c() +

labs(title = "Annual fluctuation in N. hemisphere sea ice extent",

subtitle = "Day of year on x axis",

x = "Day of year",

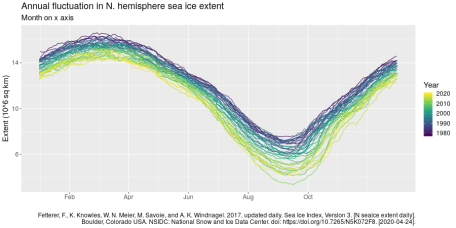
y = "Extent (10^6 sq km)",

colour = "Year",

caption = ice\_cite) +

theme(text = element\_text(size = 15))

The above looks fine. Perfect for exploratory data analysis. We can quickly see an annual pattern. However, other viewers may wish or expect to see month labels on the x axis. We can do this by setting up a tibble with values we’d like on the axis. With this in place we can call this as breaks and labels for the axis (no doubt there is a fancy function way of doing this). It’s not perfect, the labels appear at the start of each month and given months have differing lengths it’s not easy to place them in the middle (one option is to use the 15th of each month). It could make sense to have variable grid line spacing, where the lines match the month breaks, but this would be awkward to implement and be unexpected to viewers!



doy = date(c("2016-02-01",

"2016-04-01",

"2016-06-01",

"2016-08-01",

"2016-10-01"))

doy = tibble(mon = month(x, label = T),

jul = yday(x))

df %>%

mutate(year = year(date),

date = yday(date)) %>%

ggplot(aes(date, extent,

group = year,

colour = year)) +

geom\_line() +

scale\_x\_continuous(breaks = doy$jul, labels = doy$mon) +

scale\_colour\_viridis\_c() +

labs(title = "Annual fluctuation in N. hemisphere sea ice extent",

subtitle = "Month on x axis",

x = "",

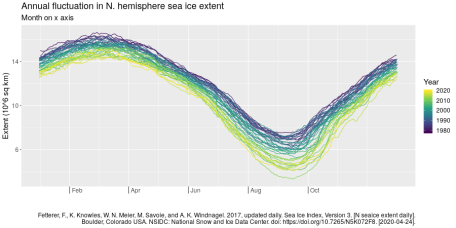
y = "Extent (10^6 sq km)",

colour = "Year",

caption = ice\_cite) +

theme(text = element\_text(size = 15))

The above is OK, but as mentioned the label position is problematic. We can solve this by hacking at the ggplot theme. We could also label the beginning of every month with this solution, but I haven’t here.



df %>%

mutate(year = year(date),

date = yday(date)) %>%

ggplot(aes(date, extent,

group = year,

colour = year)) +

geom\_line() +

scale\_x\_continuous(breaks = doy$jul, labels = doy$mon) +

scale\_colour\_viridis\_c() +

labs(title = "Annual fluctuation in N. hemisphere sea ice extent",

subtitle = "Month on x axis",

x = "",

y = "Extent (10^6 sq km)",

colour = "Year",

caption = ice\_cite) +

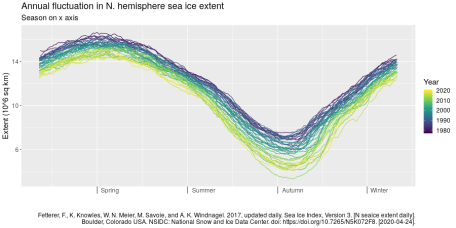
theme(text = element\_text(size = 15),

axis.ticks.length.x = unit(0.5, "cm"),

axis.text.x = element\_text(vjust = 5.5,

hjust = -0.2))

Finally, we can apply this idea to seasons:



season\_lab = tibble(jul = yday(as.Date(c("2019-03-01",

"2019-06-01",

"2019-09-01",

"2019-12-01"))),

lab = c("Spring", "Summer", "Autumn", "Winter"))

df %>%

mutate(year = year(date),

date = yday(date)) %>%

ggplot(aes(date, extent,

group = year,

colour = year)) +

geom\_line() +

scale\_x\_continuous(breaks = season\_lab$jul, labels = season\_lab$lab) +

scale\_colour\_viridis\_c() +

labs(title = "Annual fluctuation in N. hemisphere sea ice extent",

subtitle = "Season on x axis",

x = "",

y = "Extent (10^6 sq km)",

colour = "Year",

caption = ice\_cite) +

theme(text = element\_text(size = 15),

axis.ticks.length.x = unit(0.5, "cm"),

axis.text.x = element\_text(vjust = 5.5,

hjust = -0.2))

And even the below function to convert dates into seasons for fancy plotting/tables/etc.:

library(lubridate)

getSeason <- function(input.date){

numeric.date <- 100\*month(input.date)+day(input.date)

## input Seasons upper limits in the form MMDD in the "break =" option:

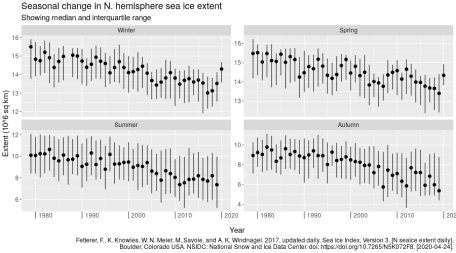
cuts <- base::cut(numeric.date, breaks = c(0,319,0620,0921,1220,1231))

# rename the resulting groups (could've been done within cut(...levels=) if "Winter" wasn't double

levels(cuts) <- c("Winter","Spring","Summer","Fall","Winter")

return(cuts)

}



season = function(in\_date){

br = yday(as.Date(c("2019-03-01",

"2019-06-01",

"2019-09-01",

"2019-12-01")))

x = yday(in\_date)

x = cut(x, breaks = c(0, br, 366))

levels(x) = c("Winter", "Spring", "Summer", "Autumn", "Winter")

x

}

df %>%

mutate(year = year(date),

sea = season(date)) %>%

group\_by(year, sea) %>%

summarise(obs = n(),

q25 = quantile(extent, 0.25),

q50 = quantile(extent, 0.5),

q75 = quantile(extent, 0.75)) %>%

filter(obs > 40) %>%

ggplot(aes(year, q50)) +

geom\_pointrange(aes(ymin = q25, ymax = q75)) +

facet\_wrap(~sea, scales = "free\_y") +

labs(title = "Seasonal change in N. hemisphere sea ice extent",

subtitle = "Showing median and interquartile range",

x = "Year",

y = "Extent (10^6 sq km)",

caption = ice\_cite) +

theme(text = element\_text(size = 15),

axis.ticks.length.x = unit(0.5, "cm"),

axis.text.x = element\_text(vjust = 5.5,

hjust = -0.2))