~~~
---
title: "glacier_exploratory"
format: html
editor: visual
---

## Load packages

```{r}
pacman::p_load('tidyverse',
               'here',
               'janitor',
               'ggthemes',
               'terra',
               'tmap',
               'patchwork',
               'stars',
               'rnaturalearth'
               )
```

## Read data

```{r}
glacier <- st_read(here('data','LIA_glacier_outlines_Alps',
'LIA_Alps_Reinthaler_Paul_2024.shp'))
```

```{r}
lake_30 <- read_delim(here('data','ViániC_2018',
                           'datasets', '1930s_glaciallakes.tab')) %>%
  clean_names()

lake_70 <- read_delim(here('data','VianiC_2018',
                           'datasets', '1970s_glaciallakes.tab'))%>%
  clean_names()

lake_80 <- read_delim(here('data','VianiC_2018',
                           'datasets', '1980s_glaciallakes.tab'))%>%
  clean_names()

lake_90 <- read_delim(here('data','VianiC_2018',
                           'datasets', '1990s_glaciallakes.tab'))%>%
  clean_names()

lake_06 <- read_delim(here('data','VianiC_2018',
                           'datasets', '2006_glaciallakes.tab'))%>%
  clean_names()


lake_12 <- read_delim(here('data','VianiC_2018',
                           'datasets', '2012_glaciallakes.tab'))%>%
  clean_names()
```

Join the `lake` datasets into one - all have the same column names so we can `bind_rows()`

```{r}
lake_joined <- bind_rows(list(lake_30, lake_70, lake_80,
                           lake_06, lake_12))

```

I want to explore these a bit more so lets find the unique values
~~~

~~~
```{r}
print(paste0('There are ', n_distinct(lake_joined$event), ' glacier lakes in the
dataset'))


```

That seems a bit high, and if you open up the table youd notice that the `event` column
has repeat rows for lakes at different sites. This is good! However for seeing the number
of lakes and comparing total size, we should have some way to aggregate the lakes. I will
separate!

```{r}
#| warning: FALSE
lake_joined <- lake_joined %>%
  separate(event, into = c("lake", NA), sep = "_") # this has the added benefit of
renaming event to 'lake'
```

What's it now?

```{r}
print(paste0('There are ', n_distinct(lake_joined$lake), ' glacier lakes in the dataset'))

```

91 is a lot to plot but
Now I want to examine 4 things graphically:

1.  How do the glacier lake areas change over time
2.  Where are the glacier lakes
3.  Where are the glaciers
4.  Where do the lakes intersect with the glaciers (aka should I drop any)

##1 How do the glacier lake areas change over time


```{r}
# First we aggregate by glacier lake and year and take a sum of the total area
lake_joined %>%
  group_by(lake, date_time) %>%
  summarise(total_area = sum(perimeter_m)) %>%
  ggplot(aes(x = date_time, y = total_area,
             color = lake)) +
  geom_point()+
  geom_line()+
  theme(legend.position = 'none')
```
That is unintelligable for now, but I can see some variation.

## 2.  Where are the glacier lakes
This is going to require I make the glacier lakes have shape. To do that i'm going to make
points at each lake and add a buffer of the outline of the lakes.

```{r}
lake_geo <- lake_joined %>%
  st_as_sf(coords = c("longitude", "latitude"), crs = 4326) %>%
  st_transform( crs = 3857) %>%
  st_buffer( dist = lake_joined$perimeter_m) %>%
  st_transform(crs = 4326)

# now we can plot
~~~

```
tm_shape(lake_geo)+
  tm_polygons(fill = 'lake')+
tm_basemap()
```

It looks like our lakes are all over, lets plot out the glaciers and see their extent

```{r}
tm_shape(glacier)+
  tm_polygons(fill = 'LIA_ID')+
tm_basemap()
```

I want to crop to the shape of the aosta valley as we have a lot of glaciers and lakes there

```{r}
italy_map <- ne_states(country = "Italy", returnclass = "sf")

aosta_map <- italy_map %>%
  select(province = name, region, geometry) %>%
  filter(region == "Valle d'Aosta")
```

Now lets crop to that region and plot again

```{r}
sf::sf_use_s2(FALSE)


glacier <- glacier %>%
  st_transform(crs(aosta_map))

glacier_cropped <- st_crop(glacier, st_bbox(aosta_map))

lake_geo <- lake_geo %>%
  st_transform(crs(aosta_map))

lake_cropped <- st_crop(lake_geo, st_bbox(aosta_map))
```

now lets make a big aosta plot

```{r}
tm_basemap()+
   tm_tiles(c(CartoDB = "CartoDB.PositronOnlyLabels")) +
tm_shape(aosta_map)+
  tm_borders()+
tm_shape(glacier_cropped)+
  tm_polygons(fill = 'LIA_ID')+
tm_shape(lake_cropped)+
  tm_polygons(fill = 'lake')
```