

# Data Science 403 Project Update 1

```
In [1]: #import statements
import earthaccess
import geopandas as gpd
import pandas as pd
import matplotlib.pyplot as plt
import matplotlib.ticker as ticker
import numpy as np
import os
import random
import rasterio
import xarray as xr
import rioarray
from shapely import wkt
```

## Data Aquisition

### BedMachine Antarctica

```
In [2]: #Log into Earthdata to get dataset
earthaccess.login()
#find bedmachine on earthdata
path= earthaccess.search_data(doi='10.5067/FPSU0V1MWUB6', cloud_hosted=True, bounding_b
#open data granules
file=earthaccess.open(path)
```

Granules found: 1

Opening 1 granules, approx size: 0.79 GB

QUEUEING TASKS | : 0%| | 0/1 [00:00<?, ?it/s]

PROCESSING TASKS | : 0%| | 0/1 [00:00<?, ?it/s]

COLLECTING RESULTS | : 0%| | 0/1 [00:00<?, ?it/s]





BedMachine data can be accessed via [this link](#), and you are able to create an Earthdata login for free. The features of the data I plan to use are Firn, Surface, Thickness and bed. A more indepth review of this dataset can be found [here](#). There are no limits on how I use this data, but as we will see in the preprocessing stage, the data does not span the entire Antarctica continent.

```
In [3]: #open dataset, its a netCDF, so we first read it into an xarray
ds=xr.open_dataset(file[0],engine='h5netcdf')
ds
```





















Out[3]: xarray.Dataset

► Dimensions: (x: 13333, y: 13333)

▼ Coordinates:

<b>x</b>	(x)	int32	-3333000 -3332500 ... 3333000	 
<b>y</b>	(y)	int32	3333000 3332500 ... -3333000	 

▼ Data variables:

mapping	()	S1	...	 
mask	(y, x)	int8	...	 
firn	(y, x)	float32	...	 
surface	(y, x)	float32	...	 
thickness	(y, x)	float32	...	 
bed	(y, x)	float32	...	 
errbed	(y, x)	float32	...	 
source	(y, x)	int8	...	 
dataid	(y, x)	int8	...	 
geoid	(y, x)	int16	...	 

► Indexes: (2)

► Attributes: (17)

```
In [4]: #convert to dataframe
bedMach=ds.to_dataframe()
bedMach=bedMach.reset_index()
bedMach
```

Out[4]:

	x	y	mapping	mask	firn	surface	thickness	bed	errber
0	-3333000	3333000	b"	0	0.0	0.0	0.0	-5915.544434	Na
1	-3333000	3332500	b"	0	0.0	0.0	0.0	-5911.253418	Na
2	-3333000	3332000	b"	0	0.0	0.0	0.0	-5907.299805	Na
3	-3333000	3331500	b"	0	0.0	0.0	0.0	-5903.499512	Na
4	-3333000	3331000	b"	0	0.0	0.0	0.0	-5899.804688	Na
...	...	...	...	...	...	...	...	...	...
177768884	3333000	-3331000	b"	0	0.0	0.0	0.0	-3663.762451	Na
177768885	3333000	-3331500	b"	0	0.0	0.0	0.0	-3664.628418	Na
177768886	3333000	-3332000	b"	0	0.0	0.0	0.0	-3665.332764	Na
177768887	3333000	-3332500	b"	0	0.0	0.0	0.0	-3665.390381	Na
177768888	3333000	-3333000	b"	0	0.0	0.0	0.0	-3664.379883	Na

177768889 rows × 12 columns



## Grounding Line

In [5]:

```
#get grounding line, where ice transitions from ground to floating ice
groundLine=gpd.read_file("Antarctica_masks\scripps_antarctica_polygons_v1.shp")
landice = groundLine[groundLine['Id_text'] == 'Grounded ice or land']
```

The grounding line is where land ice stops and floating ice begins, we can think of it as the coastal line of Antarctica. The data can be accessed at [this link](#), from the Scripps Institution of Oceanography. Subglacial lakes don't exist under floating ice, so we are interested in land ice areas, thus we need to clip our datasets to the grounding line.

## Confirmed Subglacial Lakes

In [6]:

```
#get subglacial lake outlines, this data is downloaded from my research advisor's code
df=pd.read_csv('outlines.csv')
df['geometry'] = df['geometry'].apply(wkt.loads)
outlines=gpd.GeoDataFrame(df)
```

I obtained this csv from code written by my research advisor Wilson Sauthoff, who contributed to a [Github repository](#) that pulls from a paper (Siegfried & Fricker 2021). I imported directly from a csv instead of importing the code from the github because it is cleaner and is better suited for the scope of this project. If needed I can share this csv with you. The data contains the areas, represented by polygons, of confirmed subglacial lakes. I will use this data as one part of my 'target' data.

## Not Subglacial Lakes

```
In [7]: #this data also came from my research advisors code
df = pd.read_csv('nonlakePoint.csv', header=None)
coordArray=df.to_numpy()
d=pd.read_csv('nonlakePTS.csv')
nonlakeGDF= gpd.GeoDataFrame(d)
```


Similar to the confirmed lakes, the non-subglacial lakes csv was obtained from code written by Wilson Sauthoff. The csv contains x, y coordinates of places we know there is not a subglacial lake. These coordinates were created/found by taking a simple random sample of Antarctica with areas of confirmed and suspected lakes removed. This data will be used as one part of my 'target' data.

## Data Preprocessing

```
In [8]: bedMach.describe()
```

```
Out[8]:
```

	x	y	mask	firn	surface	thickness
count	1.777689e+08	1.777689e+08	1.777689e+08	1.777689e+08	1.777689e+08	1.777689e+08
mean	0.000000e+00	0.000000e+00	6.449904e-01	7.101744e+00	5.955841e+02	5.885281e+02
std	1.924453e+06	1.924453e+06	9.887093e-01	9.833180e+00	9.618719e+02	9.823881e+02
min	-3.333000e+06	-3.333000e+06	0.000000e+00	0.000000e+00	0.000000e+00	0.000000e+00
25%	-1.666500e+06	-1.666500e+06	0.000000e+00	0.000000e+00	0.000000e+00	0.000000e+00
50%	0.000000e+00	0.000000e+00	0.000000e+00	0.000000e+00	0.000000e+00	0.000000e+00
75%	1.666500e+06	1.666500e+06	2.000000e+00	1.680822e+01	5.417394e+02	6.871041e+02
max	3.333000e+06	3.333000e+06	4.000000e+00	5.092118e+01	4.818156e+03	4.822795e+03

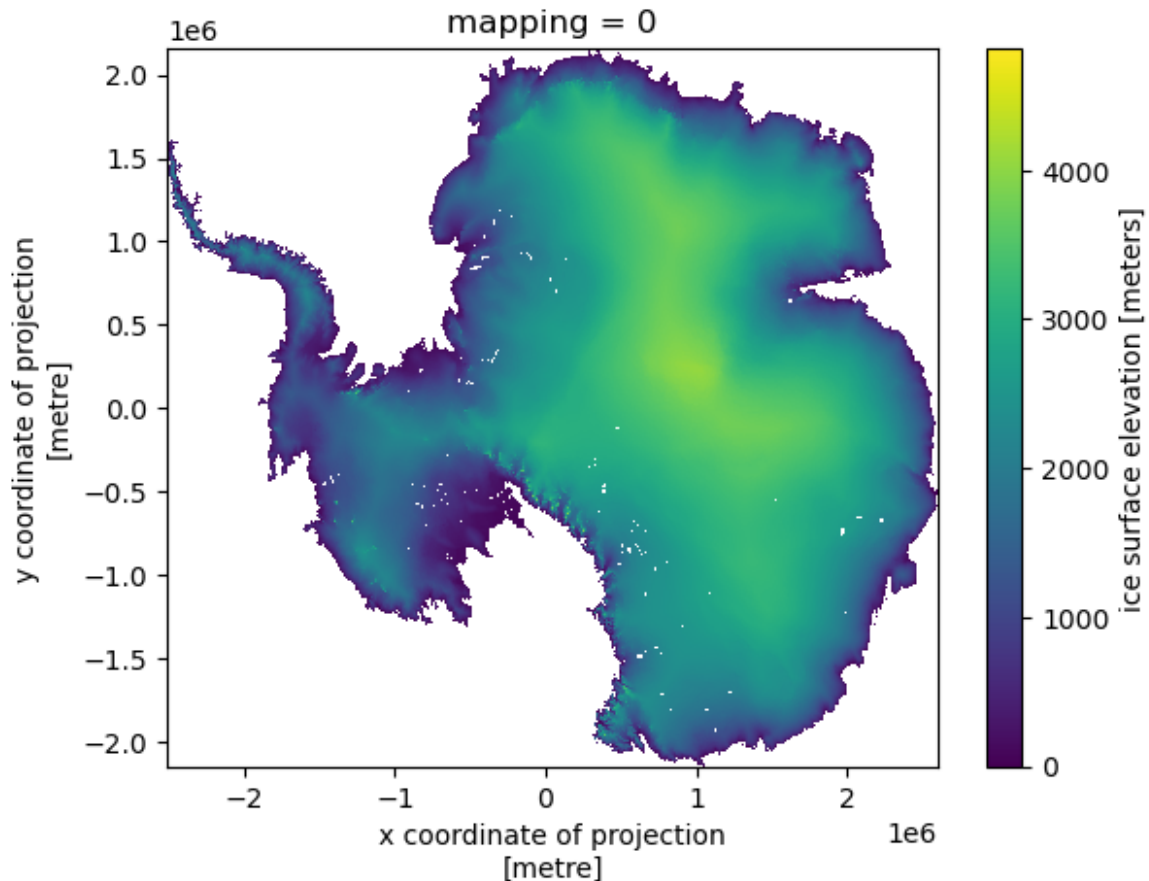


We see that for the firn, surface, and thickness, alot of values are 0. This could be due to lack of data. The data in this set is derived from various flight paths over Antarctica, but due to weather conditions and fuel limitations, there are some geographic regions that do not have any data.

```
In [9]: #clip data to grounding line, clip to coast line
ds.rio.write_crs("epsg:3031", inplace=True)
nonlake = ds.surface.rio.clip(landice.geometry.values, landice.crs)
```

```
In [10]: #image of where confirmed lakes are, we will use this as our 'target'
lakes = nonlake.rio.clip(outlines.geometry.values, outlines.crs, invert=True)
lakes.plot()
```

```
Out[10]: <matplotlib.collections.QuadMesh at 0x19aba762290>
```



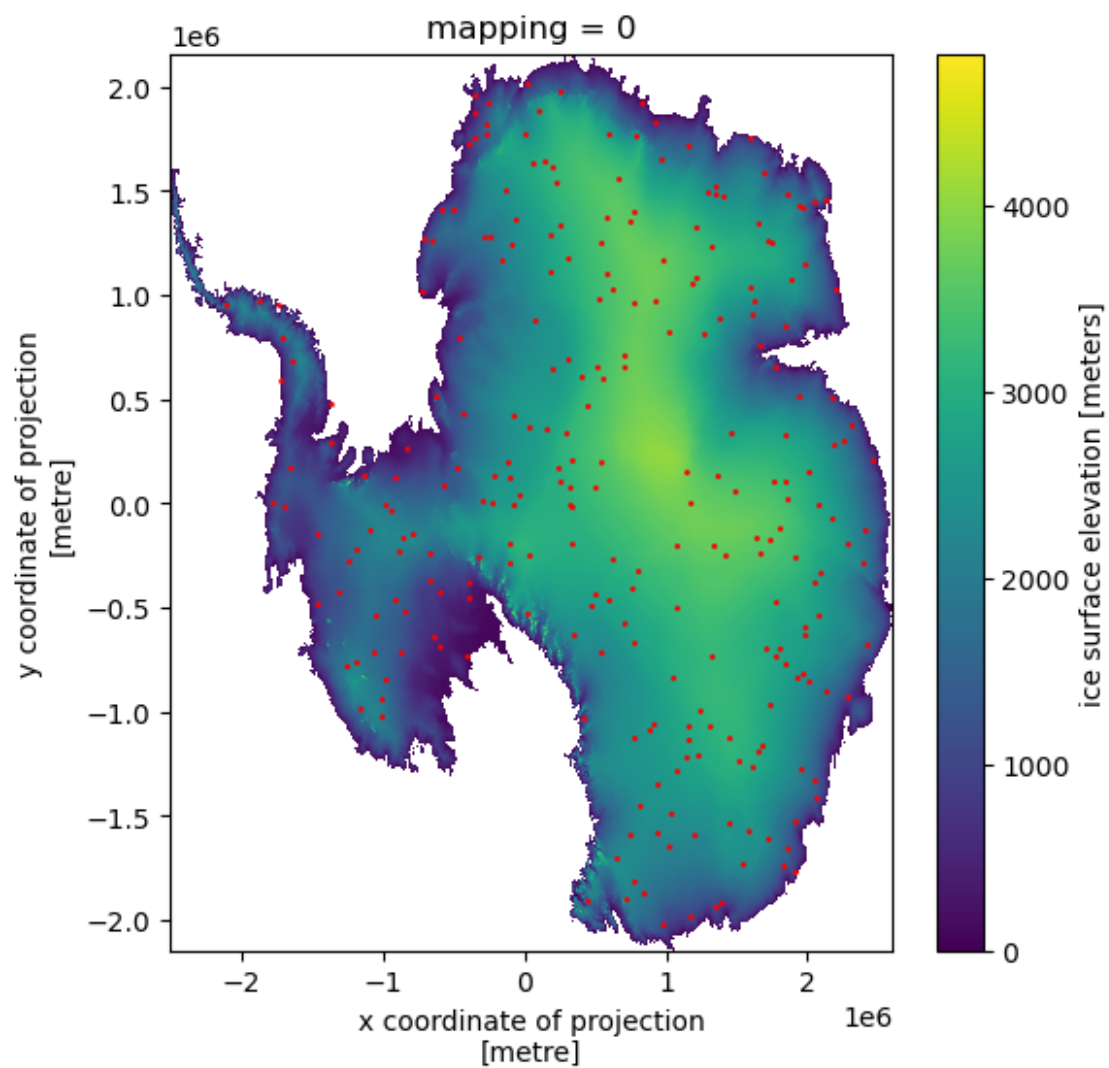
```
In [11]: #remove unnecessary columns from our outlines gpd
outlines=outlines['geometry']
outlines
```

```
Out[11]: 0      POLYGON ((-792264.327 -691480.857, -791281.458...
1      POLYGON ((-842788.063 -708464.240, -842354.948...
2      POLYGON ((-874893.221 -654533.044, -876415.673...
3      POLYGON ((-828821.778 -584874.415, -828822.032...
4      POLYGON ((-858067.460 -573467.564, -858714.391...
...
126     POLYGON ((-451544.869 -488823.261, -451209.964...
127     POLYGON ((-543163.376 -500759.165, -542800.367...
128     POLYGON ((-654478.748 -281124.560, -653777.327...
129     POLYGON ((2214185.180 -666018.604, 2214317.389...
130     POLYGON ((1985649.483 -1222665.850, 1986964.16...
Name: geometry, Length: 131, dtype: geometry
```

Now we have areas where we know a subglacial lake exists. Now we need areas where subglacial lakes do not exist.

```
In [12]: #image of where confirmed nonlakes are, we will use this as our 'target'
plt.figure(figsize=(6, 6))
nonlake.plot()
x, y = zip(*coordArray)
plt.scatter(x, y, s=1, color='red', marker='o')
```

```
Out[12]: <matplotlib.collections.PathCollection at 0x19a4d6def90>
```



```
In [13]: #remove unnecessary columns from our nonlakes gpd
nonlakeGDF=nonlakeGDF.drop(columns=['name'])
nonlakeGDF
```

Out[13]:

	geometry	type
<b>0</b>	POINT (1392000 -34500)	nonlake
<b>1</b>	POINT (1175000 393000)	nonlake
<b>2</b>	POINT (1746500 -1128000)	nonlake
<b>3</b>	POINT (705500 -1992000)	nonlake
<b>4</b>	POINT (1326000 -1626500)	nonlake
...	...	...
<b>257</b>	POINT (2095000 -985000)	nonlake
<b>258</b>	POINT (1837000 -204500)	nonlake
<b>259</b>	POINT (377500 1067500)	nonlake
<b>260</b>	POINT (892500 -2042500)	nonlake
<b>261</b>	POINT (407500 -317000)	nonlake

262 rows × 2 columns

## Data processing still to do:

1. Combine lake and nonlake geodf, add another column with 'target type'.
2. Create accumulation flow
3. Get our bedmachine values at the center of our lake and nonlake values, by finding the centroid of our polygons in x, y coordinates.

Then we should have what we need to create the model

## Ethics

My project does not have any clear ethical or societal impacts because it is located in an area of the world where very few humans reside, and those that do live there are not permanent residents, instead they are mainly researchers and military personnel. However, there are two main impacts we can consider, the impact subglacial lakes have on climate change, and what we can do with knowing where these subglacial lakes are.

If the glaciers in Antarctica and Greenland were to all melt, sea level would rise by approximately 210 feet. This would cause a large increase in coastal erosion and an increase in storm surges, as warming air and water temperatures contribute to more frequent storms, such as hurricanes. So understanding any and all causes of glacial melt, is very important. The faster glaciers flow/move the more frictional heating we have, which leads to more ice melt. The flow of glaciers is extremely dependent on their subglacial hydrology, including subglacial lakes. So if we know exactly where subglacial lakes are, and what geophysical characteristics and features represent them, we can better track them and potentially visit them and collect other valuable data. While

this information would not help slow or fix climate change, what it can do is allow us to better track how quickly ice is melting, and what climate change is doing to the ice sheets.

Additionally, because places like Antarctica are so cold, they are able to preserve information about the past very well. Subglacial lakes contain information on ancient climate, and ancient microorganisms, who have been able to survive for over thousands of years. Similar to how ice cores provide information on how greenhouse gasses have changed throughout the past, we can look at the water in these lakes to gain vital information on atmospheric conditions.

While neither of these considerations directly impact our daily lives, they can provide scientists with vital information on how climate change is impacting the world, and what that might mean for our future.