Eric_Keenan_HW1

February 7, 2020

1 Ice Sheets and Climate - Eric Keenan - Homework # 1

```
In [1]: import numpy as np
        import pandas as pd
        import matplotlib.pyplot as plt
```

2 1. Marine Records

1.b. Benthic δ^{18}O has generally been increasing over the last 70 million years. This means that ice volumne has been increasing. As ice sheets grow, δ^{18}O depleted water accumulates on ice sheets which means that δ^{18}O enriched water accumulates in the ocean.

1.c. 34 million years ago there was a sudden increase in δ^{18}O, this indicates a cooling of the climate. Therefore it is likely that 34 million years ago, ice sheets developed and grew on Earth leading to the observed increase in δ^{18}O in the marine record.

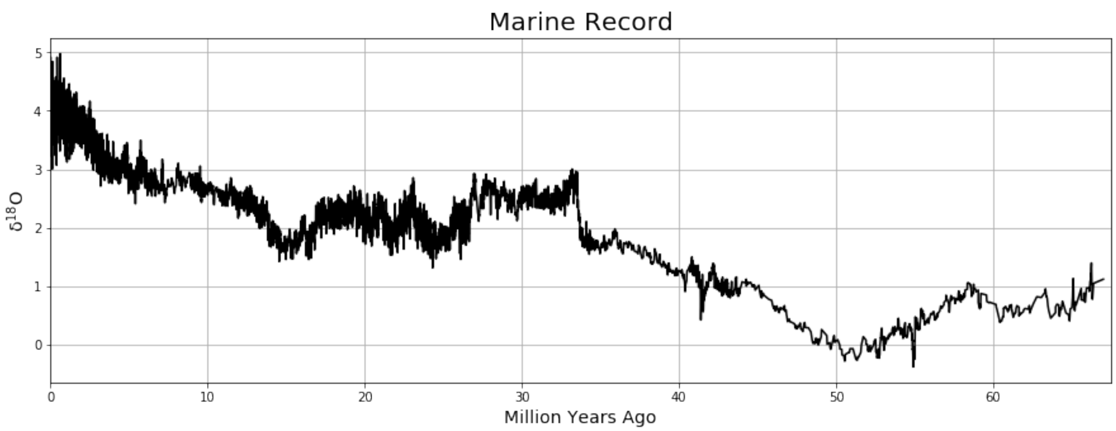
1.d. Milankovitch cycles arising from variations in Earth's orbital paramters explain the high frequency variations. Meanwhile, slower tectonic processes manipulate Earth's climate and therefore δ^{18}O over longer time scales.

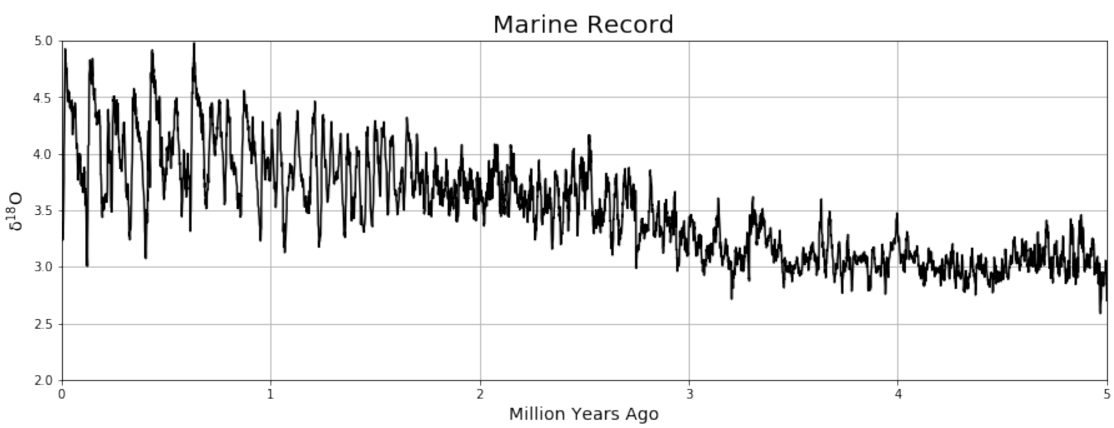
```
In [2]: # Load data
        file_path="zachos2001.csv"
        zachos = pd.read_csv(file_path)
        marine_time = zachos['Age (Ma)']
        marine_d18O = zachos['d18O(5pt**)']

        # Plot last ~70 million years
        fig1 = plt.figure(1, figsize=(15,5))
        plt.plot(marine_time, marine_d18O, 'k')
        plt.xlabel("Million Years Ago", fontsize=14)
        plt.ylabel("\u03B4$^{18}$O", fontsize=14)
        plt.title("Marine Record", fontsize=20)
        plt.xlim([0, np.nanmax(marine_time)])
        plt.grid()

        # Plot last 5 million years
```

```
fig1 = plt.figure(2, figsize=(15,5))
plt.plot(marine_time, marine_d18O, 'k')
plt.xlabel("Million Years Ago", fontsize=14)
plt.ylabel("\u03B4$^{18}$O", fontsize=14)
plt.title("Marine Record", fontsize=20)
plt.xlim([0, 5])
plt.ylim([2, 5])
plt.grid()
```





3 2. Ice Core Records

2.c. Glacial periods are relatively cool with low δ^{18}O, while interglacials are relatively warm with high δ^{18}O. Glacial periods are typically 2-4 times longer than interglacial periods. We are currently in an interglacial period, the Holocene.

2.d. Both records show warming in the lat 20,000 years leading to the Holocene. However, the GISP-2 record shows more variability in δ^{18}O than Dome Fuji. GISP-2 has higher temperal resolution δ^{18}O in the last 40,000 years, while Dome Fuji appears to have higher temporal resolution before 60,000 years ago.

2.e. Both records indicate consistent timing glacial and interglacial periods. However, Dome Fuji exhibits larger δ^{18}O variability than the marine core, perhaps because the ocean is a larger water resevoir than Antarctica. Additionally, the ice core is a much higher temperal resolution than the marine core.

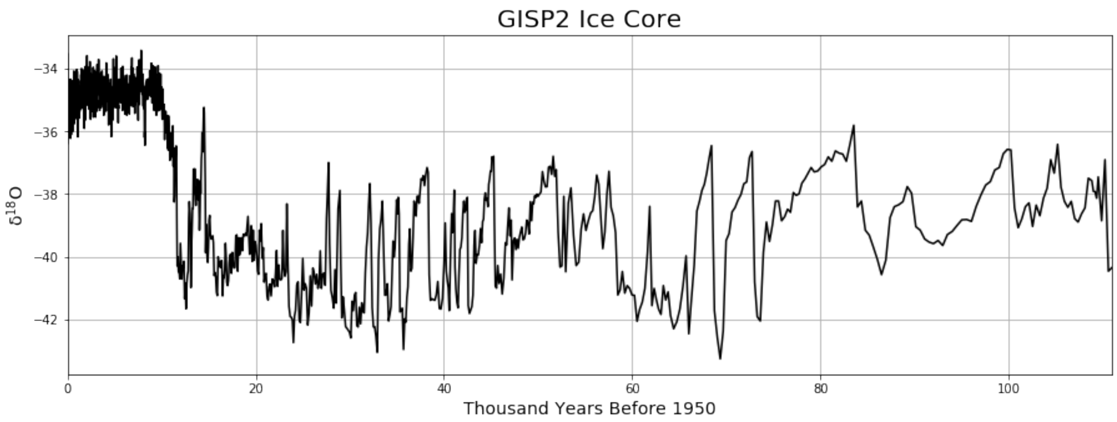
2.f. Accumulation rates must be low enough such that up 1.5 million years of accumulation are present in the ice column. Temperatures must be low enough in order to prevent melt and sublimation from ablating the ice. And ice flow must be minimal so that scientists can be confident that the ice in the core originatated from the drilling location as opposed to advecting there.

3.1 GISP2

In [3]:
```
# Load data
file_path="gispd18o.txt"
gisp = np.loadtxt(file_path, skiprows=51)
gisp2_depth = gisp[:,0]
gisp2_d18O = gisp[:,1]
gisp2_age = gisp[:,2]

# Filter data
gisp2_age[gisp2_age > 200000] = np.nan
gisp2_age[gisp2_d18O > 10] = np.nan
gisp2_d18O[gisp2_age > 200000] = np.nan
gisp2_d18O[gisp2_d18O > 10] = np.nan

# Plot
fig3 = plt.figure(3, figsize=(15,5))
plt.plot(gisp2_age / 1000, gisp2_d18O, 'k')
plt.xlabel("Thousand Years Before 1950", fontsize=14)
plt.ylabel("\u03B4$^{18}$O", fontsize=14)
plt.title("GISP2 Ice Core", fontsize=20)
plt.xlim([0, np.nanmax(gisp2_age) / 1000])
plt.grid()
```



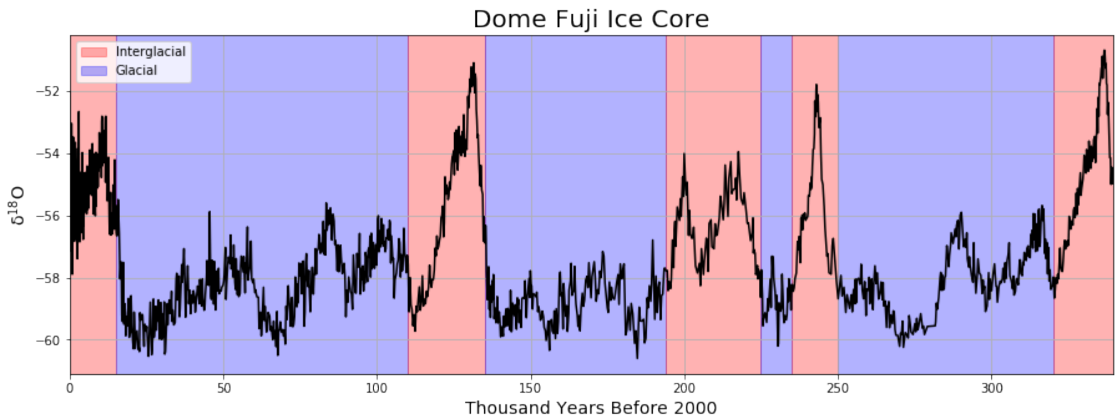
3.2 Dome Fuji

In [4]:
```
# Load data
file_path="df2012isotope-temperature.txt"
df = np.loadtxt(file_path, skiprows=1, usecols=(1,2,3,4,5,6,7))
df_age = df[:,3]
df_d180 = df[:,4]

# Plot
fig4, ax = plt.subplots(figsize=(15,5))
plt.plot(df_age, df_d180, 'k')
plt.xlabel("Thousand Years Before 2000", fontsize=14)
plt.ylabel("\u03B4$^{18}$O", fontsize=14)
plt.title("Dome Fuji Ice Core", fontsize=20)
plt.grid()
plt.xlim([0, np.nanmax(df_age)])

# Add shading for interglacial/glacial periods
ax.axvspan(0, 15, alpha=0.3, color='red', label = "Interglacial")
ax.axvspan(15, 110, alpha=0.3, color='blue', label = "Glacial")
ax.axvspan(110, 135, alpha=0.3, color='red')
ax.axvspan(135, 194, alpha=0.3, color='blue')
ax.axvspan(194, 225, alpha=0.3, color='red')
ax.axvspan(225, 235, alpha=0.3, color='blue')
ax.axvspan(235, 250, alpha=0.3, color='red')
ax.axvspan(250, 320, alpha=0.3, color='blue')
ax.axvspan(320, 340, alpha=0.3, color='red')
plt.legend(loc='upper left')
```

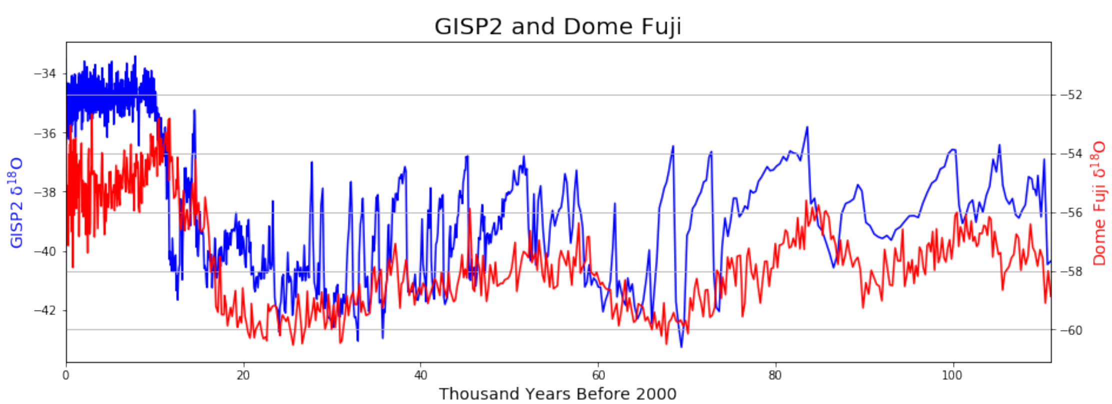
Out[4]:
```
<matplotlib.legend.Legend at 0x1170fc8d0>
```



3.3 GISP2 and Dome Fuji

```
In [5]: # Data manipulations, convert to thousands of years before 2000
        gisp2_age_yb2000 = (gisp2_age + 50) / 1000
        df_age_yb2000 = df_age
        gisp2_max = np.nanmax(gisp2_age_yb2000)

        # Plot
        fig5, ax1 = plt.subplots(figsize=(15,5))
        ax2 = ax1.twinx()
        ax1.plot(gisp2_age_yb2000, gisp2_d18O, 'b')
        ax2.plot(df_age_yb2000, df_d18O, 'r')
        ax1.set_xlabel("Thousand Years Before 2000", fontsize=14)
        ax1.set_ylabel("GISP2 \u03B4$^{18}$O", fontsize=14, color='b')
        ax2.set_ylabel("Dome Fuji \u03B4$^{18}$O", fontsize=14, color='r')
        plt.title("GISP2 and Dome Fuji", fontsize=20)
        plt.xlim([0, gisp2_max])
        plt.grid()
```



3.4 Ice Cores and Marine Cores

In [6]:
```
# Data manipulations, convert to thousands of years ago
df_age_yb2000 = df_age
marine_time_yb2000 = marine_time * 1000
df_max = np.nanmax(df_age_yb2000)

# Plot
fig6, ax1 = plt.subplots(figsize=(15,5))
ax2 = ax1.twinx()
ax1.plot(marine_time_yb2000, marine_d18O, 'b')
ax2.plot(df_age_yb2000, df_d18O, 'r')
ax1.set_xlabel("Thousand Years Ago", fontsize=14)
ax1.set_ylabel("Marine \u03B4$^{18}$O", fontsize=14, color='b')
ax2.set_ylabel("Dome Fuji \u03B4$^{18}$O", fontsize=14, color='r')
plt.title("Marine Core and Dome Fuji Ice Core", fontsize=20)
plt.xlim([0, df_max])
plt.grid()
```