Programming_assignment

December 4, 2020

0.1 Introduction

This report analyzes δ^{18}O data for historical trends and meaningful statistics. δ^{18}O is a commonly used proxy for paleotemperature.

[ ]:
```
%matplotlib inline
```

[2]:
```
#Import relevant packages
import numpy as np
import matplotlib.pyplot as plt
from scipy import stats
import pandas as pd
```

0.2 Data

This code efficiently loads the data for each site, and assigns variable names to the ages and O-18 ratios.

[3]:
```
#Initialize an empty numpy array
data=np.empty([0,2])
#Load the data using a loop to write the filename, and append to the empty array
files=['1218.','577.','588.','659.','865.']
for i in files:
    for j in [1,2,3,4]:
        data=np.append(data,np.loadtxt('site'+i+str(j)+'.
↪txt',skiprows=1),axis=0)
```

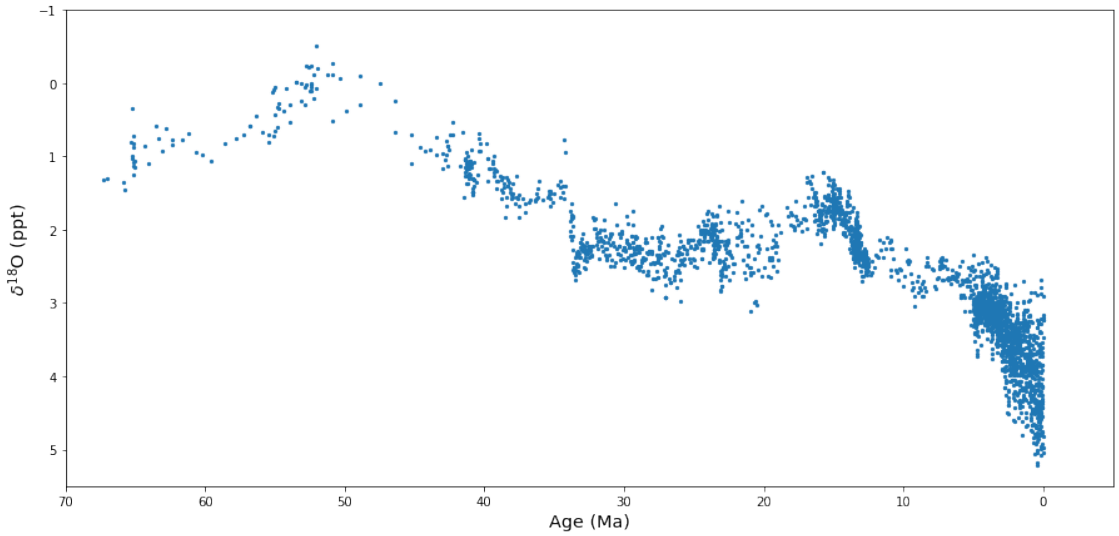
[4]:
```
#Assign variable names for convenience later
ages=data[:,0]
d18O=data[:,1]
```

0.3 Initial plot

A plot of oxygen isotope ratios and age, for the last 70 million years. As is convention, the oxygen isotope axis has been flipped, so that the graph reflects trends in temperature, which are opposite the trends in δ^{18}O. It shows a broad increase in δ^{18}O, with variation within that trend.

[22]:
```
#Plot the d18O data against age
f,ax=plt.subplots(figsize=(15,7))
ax.scatter(ages,d18O,s=5,marker='o')
ax.set_xlim(70,-5) #Display age from left to right
ax.set_ylim(5.5,-1) #Invert d18O axis as is conventional
ax.set_xlabel('Age (Ma)',fontsize='x-large')
ax.set_ylabel('$\delta^{18}$O (ppt)',fontsize='x-large')
```

[22]:
```
Text(0, 0.5, '$\\delta^{18}$O (ppt)')
```



0.4 Secondary axis

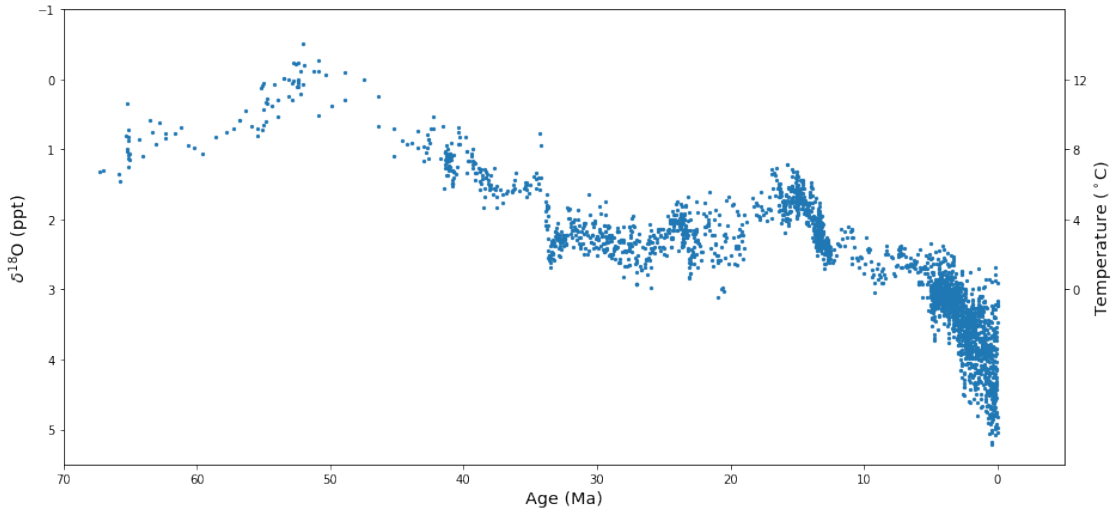
This graph adds a secondary axis to reflect temperature in Celsius between 0 to 12 degrees, as instructed, for which δ^{18}O is a proxy. The plot shows a decrease in temperature over the last 70 million years.

[23]:
```
#Recreate original plot
f,ax=plt.subplots(figsize=(15,7))
ax.scatter(ages,d18O,s=5,marker='o')
ax.set_xlim(70,-5) #Display age from left to right
ax.set_ylim(5.5,-1) #Invert d18O axis as is conventional
ax.set_xlabel('Age (Ma)',fontsize='x-large')
ax.set_ylabel('$\delta^{18}$O (ppt)',fontsize='x-large')

#Add secondary y-axis for temperature
#Start by defining required function and its inverse
def temp(x):
    return 12-4*x
```

```
def inv(temp):
    return (12-temp)/4
#secondary axis
temp_ax=ax.secondary_yaxis('right',functions=(temp,inv))
temp_ax.set_yticks([0,4,8,12])
temp_ax.set_ylabel('Temperature ($^\circ$C)',fontsize='x-large')
```

[23]:
```
Text(0, 0.5, 'Temperature ($^\\circ$C)')
```



0.5 Geological epochs

This next plot shows the boundaries of each geological epoch, and labels them.

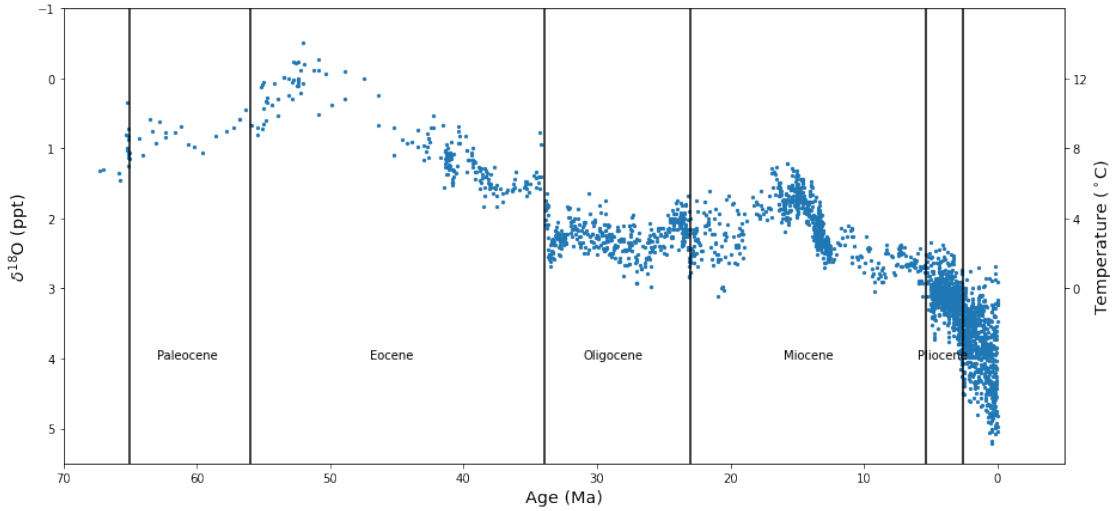
[24]:
```
#recreate original plot
f,ax=plt.subplots(figsize=(15,7))
ax.scatter(ages,d18O,s=5,marker='o')
ax.set_xlim(70,-5) #Display age from left to right
ax.set_ylim(5.5,-1) #Invert d18O axis as is conventional
ax.set_xlabel('Age (Ma)',fontsize='x-large')
ax.set_ylabel('$\delta^{18}$O (ppt)',fontsize='x-large')

#Add secondary y-axis for temperature
#Start by defining necessary function and its inverse
def temp(x):
    return 12-4*x
def inv(temp):
    return (12-temp)/4
#secondary axis
temp_ax=ax.secondary_yaxis('right',functions=(temp,inv))
```

```
temp_ax.set_yticks([0,4,8,12])
temp_ax.set_ylabel('Temperature ($^\circ$C)',fontsize='x-large')

#Mark geological epochs
epochs=[65,56,33.9,23,5.33,2.58]
for i in epochs:
    ax.plot([i,i],[5.5,-1],c='black')
#Add labels
epoch_names=['Paleocene','Eocene','Oligocene','Miocene','Pliocene']
ax.text(63,4,epoch_names[0])
ax.text(47,4,epoch_names[1])
ax.text(31,4,epoch_names[2])
ax.text(16,4,epoch_names[3])
ax.text(6,4,epoch_names[4])
```

[24]: `Text(6, 4, 'Pliocene')`



0.6 Epoch means

The mean δ^{18}O for each epoch was calculated and is shown in the table below. The means reflect the broad trend of increasing δ^{18}O, and the variance within this trend.

[8]:
```
#Calculate mean of each epoch and append to list
#Initialize an empty list
means=[]
#Select the data within the range of each epoch, and calculate the mean
for i in range(len(epochs)-1):
    #Define upper and lower bounds
    upperbound=epochs[i]
```

```
    lowerbound=epochs[i+1]
    #Create mask
    mask=(upperbound> ages) & (ages>lowerbound)
    #Select data within desired range
    values=d18O[mask]
    #Find mean of values, round, and append to empty list
    means.append(round(np.mean(values),2))
```

[14]:
```
#Create nice table of means
pd.DataFrame(means,index=epoch_names,columns=['Mean $\delta^{18}$O (ppt)'])
```

[14]:

	Mean δ^{18}O (ppt)
Paleocene	0.78
Eocene	1.01
Oligocene	2.26
Miocene	2.12
Pliocene	3.14

1 Trendline

The following plot shows a trendline for the entire data set, along with its trend. It shows δ^{18}O increasing towards the present as temperature falls. The displayed trend is the trend moving forward in time. While δ^{18}O decreases with age, it increases as it moves towards the present, so the trend is positive. The epochs have been removed for clarity.

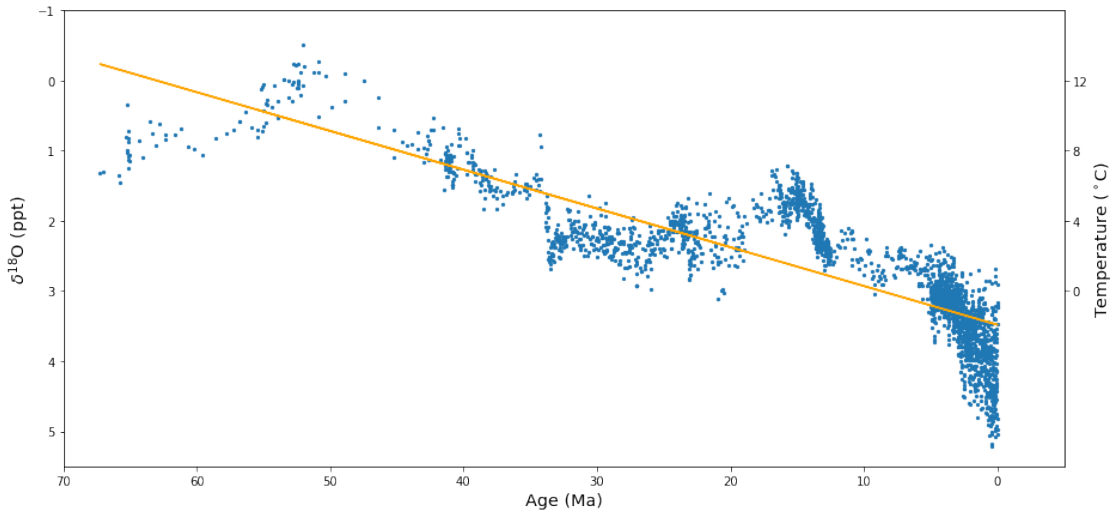
[25]:
```
f,ax=plt.subplots(figsize=(15,7))
ax.scatter(ages,d18O,s=5,marker='o')
ax.set_xlim(70,-5) #Display age from left to right
ax.set_ylim(5.5,-1) #Invert d18O axis as is conventional
ax.set_xlabel('Age (Ma)',fontsize='x-large')
ax.set_ylabel('$\delta^{18}$O (ppt)',fontsize='x-large')

#Add secondary y-axis for temperature
#Start by defining necessary function and its inverse
def temp(x):
    return 12-4*x
def inv(temp):
    return (12-temp)/4
#secondary axis
temp_ax=ax.secondary_yaxis('right',functions=(temp,inv))
temp_ax.set_yticks([0,4,8,12])
temp_ax.set_ylabel('Temperature ($^\circ$C)',fontsize='x-large')

#Calculate slope and intercept of linear best-fit line and assign to variables
overall_slope,overall_intercept,_,_,_=stats.linregress(ages,d18O)
```

```
#Plot best-fit line
x=ages
y=overall_slope*ages+overall_intercept
ax.plot(x,y,c='orange')
#Print slope
print('Trend = '+str(round(-overall_slope,1))+' ppt d18O per Ma') #the sign of␣
↪the slope is changed to reflect the trend moving forward
                                                                    #through␣
↪time
```

```
Trend = 0.1 ppt d18O per Ma
```



1.1 Trend for each epoch

This plot shows a linear trendline for each epoch, and displays the trends in a table, along with the overall trend. Again, these are the trends moving forward through time to the present. They show significant increases during the Eocene and Pliocene, and more constant levels in the Oligocene and Miocene.

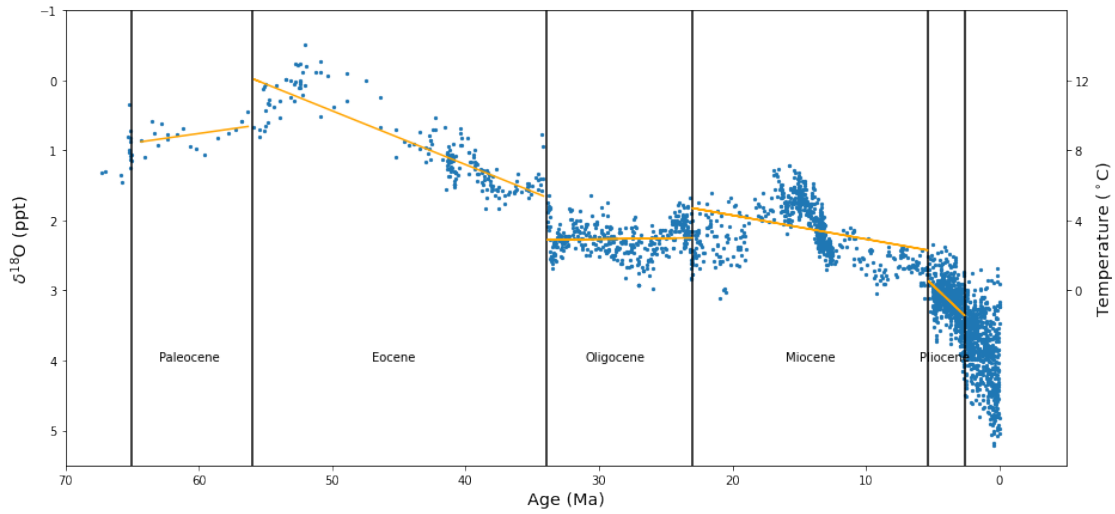
[26]:
```
#Recreate original plot
f,ax=plt.subplots(figsize=(15,7))
ax.scatter(ages,d18O,s=5,marker='o')
ax.set_xlim(70,-5) #Display age from left to right
ax.set_ylim(5.5,-1) #Invert d18O axis as is conventional
ax.set_xlabel('Age (Ma)',fontsize='x-large')
ax.set_ylabel('$\delta^{18}$O (ppt)',fontsize='x-large')

#Add secondary y-axis for temperature
#Start by defining necessary function and its inverse
```

```
def temp(x):
    return 12-4*x
def inv(temp):
    return (12-temp)/4
#secondary axis
temp_ax=ax.secondary_yaxis('right',functions=(temp,inv))
temp_ax.set_yticks([0,4,8,12])
temp_ax.set_ylabel('Temperature ($^\circ$C)',fontsize='x-large')

#Mark geological epochs
epochs=[65,56,33.9,23,5.33,2.58]
for i in epochs:
    ax.plot([i,i],[5.5,-1],c='black')
#Add labels
epoch_names=['Paleocene','Eocene','Oligocene','Miocene','Pliocene']
ax.text(63,4,epoch_names[0])
ax.text(47,4,epoch_names[1])
ax.text(31,4,epoch_names[2])
ax.text(16,4,epoch_names[3])
ax.text(6,4,epoch_names[4])

#Calculate best-fit line for each epoch, and plot it on the graph
#First initialize an empty list to store the trend for each epoch
slopes=[]
for i in range(len(epochs)-1): #a loop to do this for each epoch
    #start by slecting the data for each epoch
    #Define upper and lower bounds
    upperbound=epochs[i]
    lowerbound=epochs[i+1]
    #Create mask
    mask=(upperbound> ages) & (ages>lowerbound)
    #Select data within desired range
    x=ages[mask]
    y=d18O[mask]
    #Find slope and intercept of best fit line and plot
    m,c,_,_,_=stats.linregress(x,y)
    plt.plot(x,m*x+c,c='orange')
    #Round the slope and append it to the empty list
    slopes.append(round(-m,3))
```



[49]:
```
#Create a nice table to display each trend, along with the overall trend
pd.
 ↪DataFrame(slopes+[-round(overall_slope,3)],index=epoch_names+['Overall'],columns=['Trend␣
 ↪(ppt per Ma)'])
```

[49]:

	Trend (ppt per Ma)
Paleocene	-0.027
Eocene	0.077
Oligocene	-0.003
Miocene	0.034
Pliocene	0.182
Overall	0.055

[ ]:
```

```