SIOC 221A: HW2

In [2]:

```
import numpy as np
import matplotlib.pyplot as plt
import netCDF4
import datetime as dt
import matplotlib.dates as mdates
import pandas as pd
import xarray as xr
import scipy.stats as stats
```

In [3]:

```
# function for downloading temp data

def pier_temp_download(year):
    # years should be an array or list of (number) years of interest
    # vars_of_interest should be a string list of variables to download

    year_string = str(year)
    url_base = 'http://thredds.sccoos.org/thredds/dodsC/autoss/scripps_p
    url = url_base+year_string+'.nc'


    # read current file:
    nc = netCDF4.Dataset(url)
    # if you want the information printed as you go: print(nc)

    # only interested in time and temp
    time = nc['time'][:]
    temp = nc['temperature'][:]
    pressure = nc['pressure'][:]

    # convert to datetimes based on units of 'seconds since 1970-01-01'
    s0 = dt.datetime(1970,1,1)
    dates = [s0+dt.timedelta(seconds=float(tt)) for tt in time.data]

    return dates, temp, pressure
```

In [4]:

```
# let's take a look at 2010 and 2020

[d10, t10, p10] = pier_temp_download(2010)
[d20, t20, p20] = pier_temp_download(2020)

# check
if len(d10) != len(t10):
    print(f'lengths are not the same - {len(d10)} dates while {len(t10)}
else:
    print('2010 lengths the same!')
if len(d20) != len(t20):
    print(f'lengths are not the same - {len(d10)} dates while {len(t10)}
else:
    print('2020 lengths the same!')
```

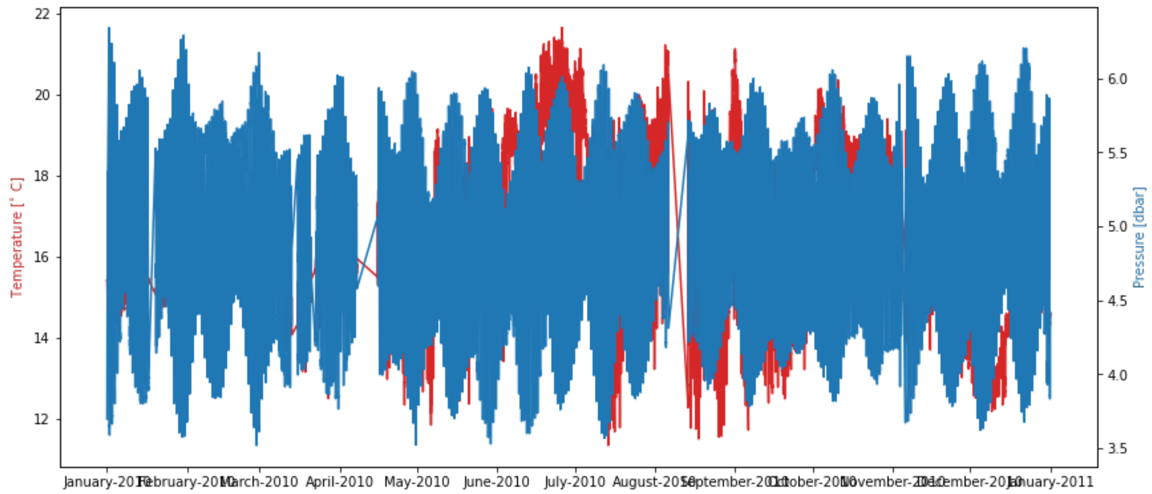
```
2010 lengths the same!
2020 lengths the same!
```

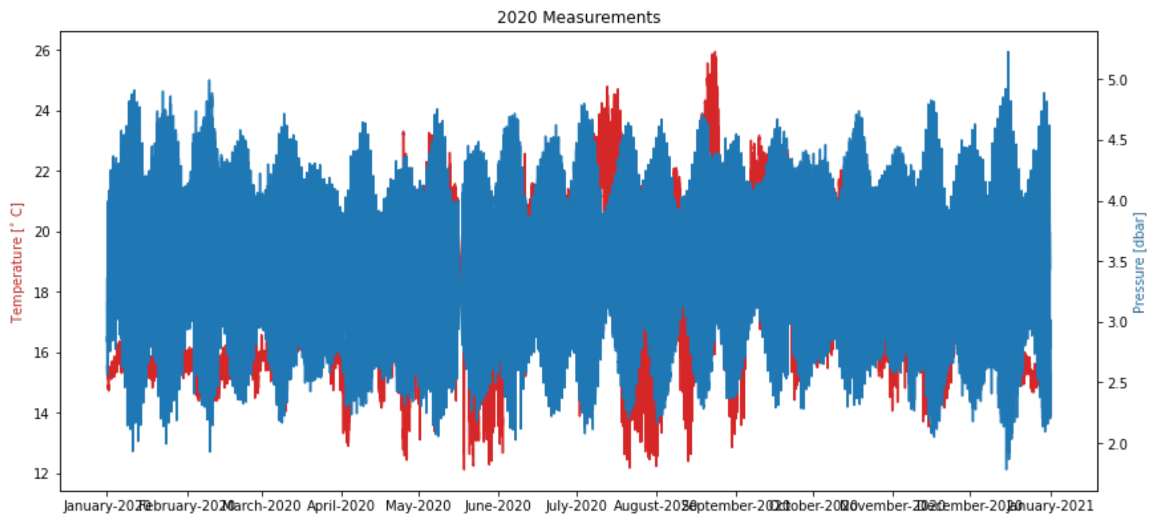
Q1: Visual Evaluation

For both temperature and pressure, plot the records from the two years. Are there obvious discrepancies between the two years? What years are covered? Any other observations?

In [5]:

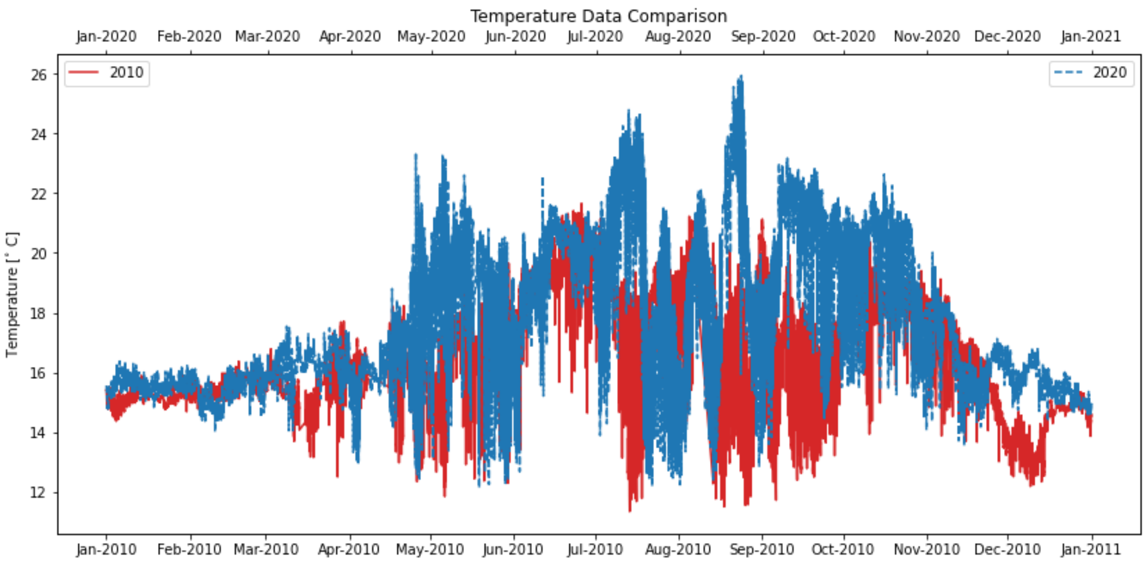
```
fig,(ax1,ax2) = plt.subplots(2,1,figsize=(14,14))
ax1.plot(d10,t10,color='tab:red')
ax1.set_ylabel(r'Temperature [$^{\circ}$ C]',color='tab:red');
ax11 = ax1.twinx()
ax11.plot(d10,p10,color='tab:blue'); ax11.set_ylabel('Pressure [dbar]',c
ax1.xaxis.set_major_formatter(mdates.DateFormatter('%B-%Y'))
ax1.xaxis.set_major_locator(mdates.MonthLocator())

ax2.plot(d20,t20,color='tab:red'); ax2.set_ylabel(r'Temperature [$^{\circ
ax22 = ax2.twinx()
ax22.plot(d20,p20,color='tab:blue'); ax22.set_ylabel('Pressure [dbar]',c
ax2.xaxis.set_major_formatter(mdates.DateFormatter('%B-%Y'))
ax2.xaxis.set_major_locator(mdates.MonthLocator())
ax2.set(title='2020 Measurements')

plt.show()
```

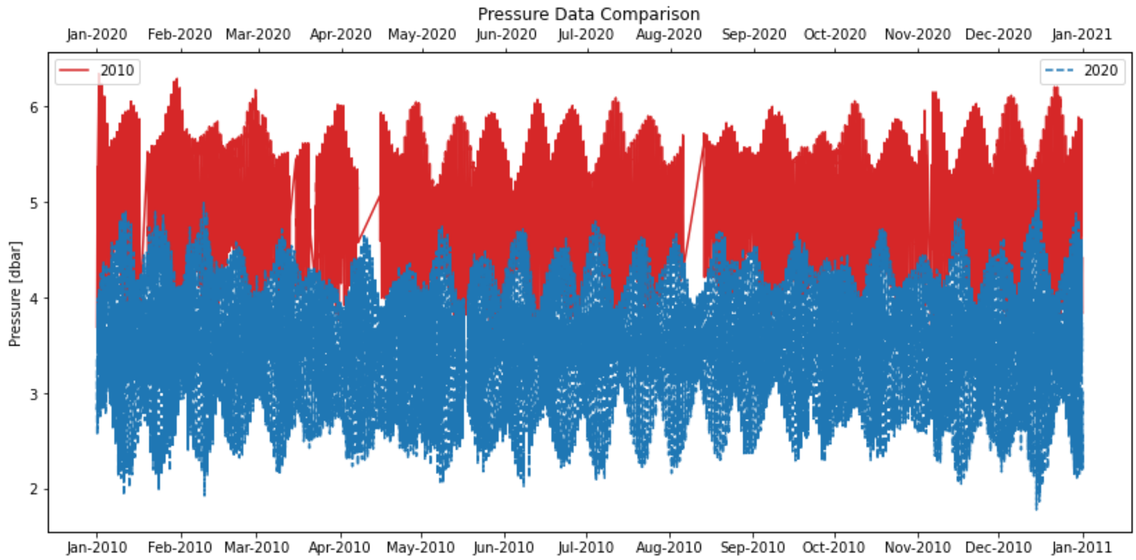




In [6]:

```
# that didn't demonstrate what i wanted to see

fig,(ax1,ax2) = plt.subplots(2,1,figsize=(14,14))
t1 = ax1.plot(d10,t10,color='tab:red',label='2010')
ax1.xaxis.set_major_formatter(mdates.DateFormatter('%b-%Y'))
ax1.xaxis.set_major_locator(mdates.MonthLocator())
ax1.legend(loc='upper left')
ax11 = ax1.twiny()
t2 = ax11.plot(d20,t20,color='tab:blue',linestyle='--',label='2020')
ax1.set_ylabel(r'Temperature [$^{\circ}$ C]');
ax11.xaxis.set_major_formatter(mdates.DateFormatter('%b-%Y'))
ax11.xaxis.set_major_locator(mdates.MonthLocator())
# ax1.legend([t1,t2],['2010','2020'])
ax11.legend(loc='upper right')
ax1.set(title='Temperature Data Comparison')

p1 = ax2.plot(d10,p10,color='tab:red',label='2010');
ax2.xaxis.set_major_formatter(mdates.DateFormatter('%b-%Y'))
ax2.xaxis.set_major_locator(mdates.MonthLocator())
ax2.legend(loc='upper left')
p2 = ax22 = ax2.twiny()
ax22.plot(d20,p20,color='tab:blue',linestyle='--',label='2020');
ax22.xaxis.set_major_formatter(mdates.DateFormatter('%b-%Y'))
ax22.xaxis.set_major_locator(mdates.MonthLocator())
ax2.set_ylabel(r'Pressure [dbar]');
ax22.legend(loc='upper right')
ax2.set(title='Pressure Data Comparison')

plt.show()
```





The compared years are 2010 and 2020. There are a few obvious discrepancies, particularly how the pressure measurements in 2010 vs 2020 have a clear shift of about 2dbar (roughly 2m). Since similar patterns are visually clear in the pressure record, I wonder if the measurement device that recorded was moved or if it was some instrument change between the time periods. Otherwise, the variation and signal in the pressure data between the years roughly match. On the other hand, while the temperature data in both years is more variable in the summer than the winter, it looks like the extreme warm temperatures are higher in the summer of 2020 whereas they are colder in the summer of 2010.

Q2: Means

For each quantity, evaluate whether the sample mean between the two years is consistent within error bars. Do you obtain different results if you subsample the data at (say) once per day?

In [13]:

```
N10 = len(t10)
mean10 = np.nanmean(t10)
std10 = np.std(t10)/np.sqrt(N10)

N20 = len(t20)
mean20 = np.nanmean(t20)
std20 = np.std(t20)
err20 = std20/np.sqrt(N20)

print(f'2010 Temperature Mean is {mean10: .4f} deg C with standard devia
print(f'2020 Temperature Mean is {mean20: .4f} deg C with standard devia

# within error bars?

# calculate myself:
# variance: sig^2 = sum( ( a_i * (xi - mu_i)^2)
# standard error of the mean: sig = sig/sqrt(N)

sum10 = 0; N10 = 0; # sum10 = sig^2 and N is number of observations
for val in t10:
    sig = abs(val-mean10)
    sig2 = np.power(sig,2)
    sum10 = sum10 + sig2
    N10 = N10 + 1


sum20 = 0; N20 = 0;
for val in t20:
    sig = abs(val-mean20)
    sig2 = np.power(sig,2)
    sum20 = sum20 + sig2
    N20 = N20 + 1


var10 = sum10/N10; sig10 = np.sqrt(var10); err10 = sig10/np.sqrt(N10)
var20 = sum20/N20; sig20 = np.sqrt(var20); err20 = sig20/np.sqrt(N20)


print(f'2010 variance = {var10: .4f} and std error of mean = {err10: .4f
print(f'2020 variance = {var20: .4f} and std error of mean = {err20: .4f
```

```
2010 Temperature Mean is  16.5155 deg C with standard deviation of  0.007
8
2020 Temperature Mean is  17.7528 deg C with standard deviation of  2.551
9
2010 variance =  3.7261 and std error of mean =  0.0078
2020 variance =  6.5122 and std error of mean =  0.0071
```

In [14]:

```
y = 0
if mean10+err10 >= mean20 and mean10-err10 <= mean20:
    # this means that 2020 mean falls inside 2010 error bars
    print('2020 within 2010 error bars')
    y = y + 1
else:
    print('2020 outside of 2010 error bars')


if mean20+err20 >= mean10 and mean20-err20 <= mean10:
    # this means that 2010 mean falls inside 2020 error bars
    print('2010 within 2020 error bars')
    y = y + 1
else:
    print('2010 outside of 2020 error bars')

if y == 2:
    print('Sample means are consistent')
elif y < 2:
    print('Sample means are not consistent')
```

```
2020 outside of 2010 error bars
2010 outside of 2020 error bars
Sample means are not consistent
```

In [15]:

```
# subsample data
## note: the way I subsample selects the first measurement of the day, a
# (early in day data saved in sampling)

subsampler = np.array([])
subdays10 = np.array([]) # this will be an array tracking the dates (1 o
sub10 = np.array([]) # this array will track the subsampled temp measure

for nn,dd in enumerate(d10):
    # loop through all dates in the data
    date_finder = dt.datetime(dd.year,dd.month,dd.day)
    if date_finder not in subsampler:
        # if encountering a new day, then will enter this loop
        subsampler = np.append(subsampler, date_finder)
        subdays10 = np.append(subdays10,dd) # saving new days in array
        sub10 = np.append(sub10,t10[nn]) # saving the temp measurement o

# calculate mean & stdev
sub10mean = np.nanmean(sub10)
sub10std = np.std(sub10)
sub10err = np.sqrt(sub10std)/np.sqrt(len(sub10))

# same thing for 2020 data subsampling

subsampler = np.array([])
subdays20 = np.array([]) # dates of subsampling
sub20 = np.array([]) # temps of subsampling

for nn,dd in enumerate(d20):
    # loop through all dates in the data
    date_finder = dt.datetime(dd.year,dd.month,dd.day)
    if date_finder not in subsampler:
        # if encountering a new day, then will enter this loop
        subsampler = np.append(subsampler, date_finder)
        subdays20 = np.append(subdays20,dd) # saving new days in array
        sub20 = np.append(sub20,t20[nn]) # saving the temp measurement o

# calculate mean, stdev, stderr
sub20mean = np.nanmean(sub20)
sub20std = np.std(sub20)
sub20err = np.sqrt(sub20std)/np.sqrt(len(sub20))
```

In [16]:

```
# check the saved subsampling is actually a subsampling:
print(subdays10)
print(subdays20)
print(f'Number of subsampled observations in 2010: {len(subdays10)}')
print(f'Number of subsampled observations in 2020: {len(subdays20)}')
```

```
[datetime.datetime(2010, 1, 1, 0, 7, 25)
 datetime.datetime(2010, 1, 2, 0, 7, 25)
 datetime.datetime(2010, 1, 3, 0, 7, 25)
 datetime.datetime(2010, 1, 4, 0, 7, 25)
 datetime.datetime(2010, 1, 5, 0, 7, 25)
 datetime.datetime(2010, 1, 6, 0, 7, 25)
 datetime.datetime(2010, 1, 7, 0, 7, 25)
 datetime.datetime(2010, 1, 8, 0, 7, 25)
 datetime.datetime(2010, 1, 9, 0, 7, 25)
 datetime.datetime(2010, 1, 10, 0, 7, 25)
 datetime.datetime(2010, 1, 11, 0, 7, 25)
 datetime.datetime(2010, 1, 12, 0, 7, 25)
 datetime.datetime(2010, 1, 13, 0, 7, 25)
 datetime.datetime(2010, 1, 14, 0, 7, 25)
 datetime.datetime(2010, 1, 15, 0, 7, 25)
 datetime.datetime(2010, 1, 16, 0, 7, 25)
 datetime.datetime(2010, 1, 17, 0, 7, 25)
 datetime.datetime(2010, 1, 19, 17, 35, 25)
 datetime.datetime(2010, 1, 20, 0, 3, 25)
 datetime.datetime(2010, 1, 21, 0, 3, 25)
 datetime.datetime(2010, 1, 22, 0, 3, 25)
 datetime.datetime(2010, 1, 23, 0, 3, 25)
 datetime.datetime(2010, 1, 24, 0, 3, 25)
 datetime.datetime(2010, 1, 25, 0, 3, 25)
 datetime.datetime(2010, 1, 26, 0, 3, 25)
 datetime.datetime(2010, 1, 27, 0, 7, 25)
 datetime.datetime(2010, 1, 28, 0, 7, 25)
 datetime.datetime(2010, 1, 29, 0, 7, 25)
 datetime.datetime(2010, 1, 30, 0, 7, 25)
 datetime.datetime(2010, 1, 31, 0, 7, 25)
 datetime.datetime(2010, 2, 1, 0, 7, 25)
 datetime.datetime(2010, 2, 2, 0, 7, 25)
 datetime.datetime(2010, 2, 3, 0, 7, 25)
 datetime.datetime(2010, 2, 4, 0, 7, 25)
 datetime.datetime(2010, 2, 5, 0, 7, 25)
 datetime.datetime(2010, 2, 6, 0, 7, 25)
 datetime.datetime(2010, 2, 7, 0, 7, 25)
 datetime.datetime(2010, 2, 8, 0, 7, 25)
 datetime.datetime(2010, 2, 9, 0, 7, 25)
 datetime.datetime(2010, 2, 10, 0, 3, 25)
 datetime.datetime(2010, 2, 11, 0, 3, 25)
 datetime.datetime(2010, 2, 12, 0, 3, 25)
 datetime.datetime(2010, 2, 13, 0, 3, 25)
 datetime.datetime(2010, 2, 14, 0, 3, 25)
 datetime.datetime(2010, 2, 15, 0, 3, 25)
 datetime.datetime(2010, 2, 16, 0, 3, 25)
 datetime.datetime(2010, 2, 17, 0, 7, 25)
 datetime.datetime(2010, 2, 18, 0, 7, 25)
 datetime.datetime(2010, 2, 19, 0, 7, 25)
 datetime.datetime(2010, 2, 20, 0, 3, 25)
 datetime.datetime(2010, 2, 21, 0, 3, 25)
 datetime.datetime(2010, 2, 22, 0, 3, 25)
 datetime.datetime(2010, 2, 23, 0, 3, 25)
 datetime.datetime(2010, 2, 24, 0, 7, 25)
 datetime.datetime(2010, 2, 25, 0, 7, 25)
 datetime.datetime(2010, 2, 26, 0, 7, 25)
 datetime.datetime(2010, 2, 27, 0, 7, 25)
 datetime.datetime(2010, 2, 28, 0, 7, 25)
 datetime.datetime(2010, 3, 1, 0, 7, 25)
 datetime.datetime(2010, 3, 2, 0, 7, 25)
 datetime.datetime(2010, 3, 3, 0, 3, 25)
 datetime.datetime(2010, 3, 4, 0, 3, 25)
 datetime.datetime(2010, 3, 5, 0, 3, 25)
```

```
datetime.datetime(2010, 3, 6, 0, 3, 25)
datetime.datetime(2010, 3, 7, 0, 3, 25)
datetime.datetime(2010, 3, 8, 0, 3, 25)
datetime.datetime(2010, 3, 9, 0, 3, 25)
datetime.datetime(2010, 3, 10, 0, 3, 25)
datetime.datetime(2010, 3, 11, 0, 3, 25)
datetime.datetime(2010, 3, 12, 0, 3, 25)
datetime.datetime(2010, 3, 13, 0, 3, 25)
datetime.datetime(2010, 3, 15, 17, 7, 25)
datetime.datetime(2010, 3, 16, 0, 3, 25)
datetime.datetime(2010, 3, 17, 0, 7, 25)
datetime.datetime(2010, 3, 18, 0, 7, 25)
datetime.datetime(2010, 3, 19, 0, 7, 25)
datetime.datetime(2010, 3, 20, 0, 7, 25)
datetime.datetime(2010, 3, 22, 15, 59, 25)
datetime.datetime(2010, 3, 23, 0, 7, 25)
datetime.datetime(2010, 3, 24, 0, 23, 15)
datetime.datetime(2010, 3, 25, 0, 7, 15)
datetime.datetime(2010, 3, 26, 0, 7, 15)
datetime.datetime(2010, 3, 27, 0, 7, 15)
datetime.datetime(2010, 3, 28, 0, 7, 15)
datetime.datetime(2010, 3, 29, 0, 7, 15)
datetime.datetime(2010, 3, 30, 0, 7, 15)
datetime.datetime(2010, 3, 31, 0, 3, 15)
datetime.datetime(2010, 4, 1, 0, 3, 15)
datetime.datetime(2010, 4, 2, 0, 3, 15)
datetime.datetime(2010, 4, 3, 0, 7, 15)
datetime.datetime(2010, 4, 4, 0, 7, 15)
datetime.datetime(2010, 4, 5, 0, 3, 15)
datetime.datetime(2010, 4, 6, 0, 3, 15)
datetime.datetime(2010, 4, 7, 0, 3, 15)
datetime.datetime(2010, 4, 15, 19, 43, 15)
datetime.datetime(2010, 4, 16, 0, 7, 15)
datetime.datetime(2010, 4, 17, 0, 7, 15)
datetime.datetime(2010, 4, 18, 0, 7, 15)
datetime.datetime(2010, 4, 19, 0, 7, 15)
datetime.datetime(2010, 4, 20, 0, 7, 15)
datetime.datetime(2010, 4, 21, 0, 7, 15)
datetime.datetime(2010, 4, 22, 0, 7, 15)
datetime.datetime(2010, 4, 23, 0, 7, 15)
datetime.datetime(2010, 4, 24, 0, 7, 15)
datetime.datetime(2010, 4, 25, 0, 7, 15)
datetime.datetime(2010, 4, 26, 0, 7, 15)
datetime.datetime(2010, 4, 27, 0, 3, 15)
datetime.datetime(2010, 4, 28, 0, 3, 15)
datetime.datetime(2010, 4, 29, 0, 3, 15)
datetime.datetime(2010, 4, 30, 0, 3, 15)
datetime.datetime(2010, 5, 1, 0, 3, 15)
datetime.datetime(2010, 5, 2, 0, 3, 15)
datetime.datetime(2010, 5, 3, 0, 3, 15)
datetime.datetime(2010, 5, 4, 0, 3, 15)
datetime.datetime(2010, 5, 5, 0, 3, 15)
datetime.datetime(2010, 5, 6, 0, 3, 15)
datetime.datetime(2010, 5, 7, 0, 3, 15)
datetime.datetime(2010, 5, 8, 0, 3, 15)
datetime.datetime(2010, 5, 9, 0, 3, 15)
datetime.datetime(2010, 5, 10, 0, 3, 15)
datetime.datetime(2010, 5, 11, 0, 3, 15)
datetime.datetime(2010, 5, 12, 0, 7, 15)
datetime.datetime(2010, 5, 13, 0, 7, 15)
datetime.datetime(2010, 5, 14, 0, 7, 15)
datetime.datetime(2010, 5, 15, 0, 7, 15)
datetime.datetime(2010, 5, 16, 0, 7, 15)
```

```
datetime.datetime(2010, 5, 17, 0, 7, 15)
datetime.datetime(2010, 5, 18, 0, 7, 15)
datetime.datetime(2010, 5, 19, 0, 3, 15)
datetime.datetime(2010, 5, 20, 0, 3, 15)
datetime.datetime(2010, 5, 21, 0, 3, 15)
datetime.datetime(2010, 5, 22, 0, 3, 15)
datetime.datetime(2010, 5, 23, 0, 3, 15)
datetime.datetime(2010, 5, 24, 0, 3, 15)
datetime.datetime(2010, 5, 25, 0, 3, 15)
datetime.datetime(2010, 5, 26, 0, 3, 15)
datetime.datetime(2010, 5, 27, 0, 7, 15)
datetime.datetime(2010, 5, 28, 0, 7, 15)
datetime.datetime(2010, 5, 29, 0, 7, 15)
datetime.datetime(2010, 5, 30, 0, 7, 15)
datetime.datetime(2010, 5, 31, 0, 7, 15)
datetime.datetime(2010, 6, 1, 0, 7, 15)
datetime.datetime(2010, 6, 2, 0, 7, 15)
datetime.datetime(2010, 6, 3, 0, 7, 15)
datetime.datetime(2010, 6, 4, 0, 7, 15)
datetime.datetime(2010, 6, 5, 0, 7, 15)
datetime.datetime(2010, 6, 6, 0, 7, 15)
datetime.datetime(2010, 6, 7, 0, 7, 15)
datetime.datetime(2010, 6, 8, 0, 7, 15)
datetime.datetime(2010, 6, 9, 0, 7, 15)
datetime.datetime(2010, 6, 10, 0, 7, 15)
datetime.datetime(2010, 6, 11, 0, 7, 15)
datetime.datetime(2010, 6, 12, 0, 7, 15)
datetime.datetime(2010, 6, 13, 0, 7, 15)
datetime.datetime(2010, 6, 14, 0, 7, 15)
datetime.datetime(2010, 6, 15, 0, 7, 15)
datetime.datetime(2010, 6, 16, 0, 3, 15)
datetime.datetime(2010, 6, 17, 0, 3, 15)
datetime.datetime(2010, 6, 18, 0, 3, 15)
datetime.datetime(2010, 6, 19, 0, 3, 15)
datetime.datetime(2010, 6, 20, 0, 3, 15)
datetime.datetime(2010, 6, 21, 0, 3, 15)
datetime.datetime(2010, 6, 22, 0, 3, 15)
datetime.datetime(2010, 6, 23, 0, 3, 15)
datetime.datetime(2010, 6, 24, 0, 3, 15)
datetime.datetime(2010, 6, 25, 0, 3, 15)
datetime.datetime(2010, 6, 26, 0, 3, 15)
datetime.datetime(2010, 6, 27, 0, 3, 15)
datetime.datetime(2010, 6, 28, 0, 3, 15)
datetime.datetime(2010, 6, 29, 0, 3, 15)
datetime.datetime(2010, 6, 30, 0, 3, 15)
datetime.datetime(2010, 7, 1, 0, 3, 15)
datetime.datetime(2010, 7, 2, 0, 7, 15)
datetime.datetime(2010, 7, 3, 0, 7, 15)
datetime.datetime(2010, 7, 4, 0, 7, 15)
datetime.datetime(2010, 7, 5, 0, 7, 15)
datetime.datetime(2010, 7, 6, 0, 7, 15)
datetime.datetime(2010, 7, 7, 0, 7, 15)
datetime.datetime(2010, 7, 8, 0, 7, 15)
datetime.datetime(2010, 7, 9, 0, 7, 15)
datetime.datetime(2010, 7, 10, 0, 7, 15)
datetime.datetime(2010, 7, 11, 0, 7, 15)
datetime.datetime(2010, 7, 12, 0, 7, 15)
datetime.datetime(2010, 7, 13, 0, 7, 15)
datetime.datetime(2010, 7, 14, 0, 3, 15)
datetime.datetime(2010, 7, 15, 0, 3, 15)
datetime.datetime(2010, 7, 16, 0, 3, 15)
datetime.datetime(2010, 7, 17, 0, 3, 15)
datetime.datetime(2010, 7, 18, 0, 3, 15)
```

```
datetime.datetime(2010, 7, 19, 0, 3, 15)
datetime.datetime(2010, 7, 20, 0, 3, 15)
datetime.datetime(2010, 7, 21, 0, 3, 15)
datetime.datetime(2010, 7, 22, 0, 3, 15)
datetime.datetime(2010, 7, 23, 0, 3, 15)
datetime.datetime(2010, 7, 24, 0, 3, 15)
datetime.datetime(2010, 7, 25, 0, 3, 15)
datetime.datetime(2010, 7, 26, 0, 3, 15)
datetime.datetime(2010, 7, 27, 0, 3, 15)
datetime.datetime(2010, 7, 28, 0, 3, 15)
datetime.datetime(2010, 7, 29, 0, 3, 15)
datetime.datetime(2010, 7, 30, 0, 3, 15)
datetime.datetime(2010, 7, 31, 0, 3, 15)
datetime.datetime(2010, 8, 1, 0, 3, 15)
datetime.datetime(2010, 8, 2, 0, 3, 15)
datetime.datetime(2010, 8, 3, 0, 3, 15)
datetime.datetime(2010, 8, 4, 0, 3, 15)
datetime.datetime(2010, 8, 5, 0, 3, 15)
datetime.datetime(2010, 8, 6, 0, 3, 15)
datetime.datetime(2010, 8, 13, 18, 20, 27)
datetime.datetime(2010, 8, 14, 0, 4, 27)
datetime.datetime(2010, 8, 15, 0, 4, 27)
datetime.datetime(2010, 8, 16, 0, 4, 27)
datetime.datetime(2010, 8, 17, 0, 4, 27)
datetime.datetime(2010, 8, 18, 0, 5, 16)
datetime.datetime(2010, 8, 19, 0, 5, 16)
datetime.datetime(2010, 8, 20, 0, 1, 16)
datetime.datetime(2010, 8, 21, 0, 1, 16)
datetime.datetime(2010, 8, 22, 0, 1, 16)
datetime.datetime(2010, 8, 23, 0, 1, 16)
datetime.datetime(2010, 8, 24, 0, 1, 16)
datetime.datetime(2010, 8, 25, 0, 1, 16)
datetime.datetime(2010, 8, 26, 0, 1, 16)
datetime.datetime(2010, 8, 27, 0, 1, 16)
datetime.datetime(2010, 8, 28, 0, 1, 16)
datetime.datetime(2010, 8, 29, 0, 1, 16)
datetime.datetime(2010, 8, 30, 0, 1, 16)
datetime.datetime(2010, 8, 31, 0, 5, 16)
datetime.datetime(2010, 9, 1, 0, 5, 16)
datetime.datetime(2010, 9, 2, 0, 1, 16)
datetime.datetime(2010, 9, 3, 0, 1, 16)
datetime.datetime(2010, 9, 4, 0, 1, 16)
datetime.datetime(2010, 9, 5, 0, 1, 16)
datetime.datetime(2010, 9, 6, 0, 1, 16)
datetime.datetime(2010, 9, 7, 0, 1, 16)
datetime.datetime(2010, 9, 8, 0, 1, 16)
datetime.datetime(2010, 9, 9, 0, 1, 16)
datetime.datetime(2010, 9, 10, 0, 1, 16)
datetime.datetime(2010, 9, 11, 0, 1, 16)
datetime.datetime(2010, 9, 12, 0, 1, 16)
datetime.datetime(2010, 9, 13, 0, 1, 16)
datetime.datetime(2010, 9, 14, 0, 1, 16)
datetime.datetime(2010, 9, 15, 0, 5, 16)
datetime.datetime(2010, 9, 16, 0, 5, 16)
datetime.datetime(2010, 9, 17, 0, 5, 16)
datetime.datetime(2010, 9, 18, 0, 5, 16)
datetime.datetime(2010, 9, 19, 0, 5, 16)
datetime.datetime(2010, 9, 20, 0, 5, 16)
datetime.datetime(2010, 9, 21, 0, 5, 16)
datetime.datetime(2010, 9, 22, 0, 5, 16)
datetime.datetime(2010, 9, 23, 0, 5, 16)
datetime.datetime(2010, 9, 24, 0, 5, 16)
datetime.datetime(2010, 9, 25, 0, 5, 16)
```

```
datetime.datetime(2010, 9, 26, 0, 5, 16)
datetime.datetime(2010, 9, 27, 0, 5, 16)
datetime.datetime(2010, 9, 28, 0, 5, 16)
datetime.datetime(2010, 9, 29, 0, 1, 16)
datetime.datetime(2010, 9, 30, 0, 1, 16)
datetime.datetime(2010, 10, 1, 0, 1, 16)
datetime.datetime(2010, 10, 2, 0, 1, 16)
datetime.datetime(2010, 10, 3, 0, 1, 16)
datetime.datetime(2010, 10, 4, 0, 1, 16)
datetime.datetime(2010, 10, 5, 0, 1, 16)
datetime.datetime(2010, 10, 6, 0, 1, 16)
datetime.datetime(2010, 10, 7, 0, 1, 16)
datetime.datetime(2010, 10, 8, 0, 1, 16)
datetime.datetime(2010, 10, 9, 0, 1, 16)
datetime.datetime(2010, 10, 10, 0, 1, 16)
datetime.datetime(2010, 10, 11, 0, 1, 16)
datetime.datetime(2010, 10, 12, 0, 1, 16)
datetime.datetime(2010, 10, 13, 0, 1, 16)
datetime.datetime(2010, 10, 14, 0, 1, 16)
datetime.datetime(2010, 10, 15, 0, 1, 16)
datetime.datetime(2010, 10, 16, 0, 1, 16)
datetime.datetime(2010, 10, 17, 0, 5, 16)
datetime.datetime(2010, 10, 18, 0, 5, 16)
datetime.datetime(2010, 10, 19, 0, 5, 16)
datetime.datetime(2010, 10, 20, 0, 5, 16)
datetime.datetime(2010, 10, 21, 0, 5, 16)
datetime.datetime(2010, 10, 22, 0, 5, 16)
datetime.datetime(2010, 10, 23, 0, 5, 16)
datetime.datetime(2010, 10, 24, 0, 5, 16)
datetime.datetime(2010, 10, 25, 0, 5, 16)
datetime.datetime(2010, 10, 26, 0, 5, 16)
datetime.datetime(2010, 10, 27, 0, 1, 16)
datetime.datetime(2010, 10, 28, 0, 1, 16)
datetime.datetime(2010, 10, 29, 0, 1, 16)
datetime.datetime(2010, 10, 30, 0, 1, 16)
datetime.datetime(2010, 10, 31, 0, 1, 16)
datetime.datetime(2010, 11, 1, 0, 1, 16)
datetime.datetime(2010, 11, 2, 0, 5, 16)
datetime.datetime(2010, 11, 3, 0, 1, 16)
datetime.datetime(2010, 11, 4, 0, 1, 16)
datetime.datetime(2010, 11, 5, 22, 33, 16)
datetime.datetime(2010, 11, 6, 0, 1, 16)
datetime.datetime(2010, 11, 7, 0, 1, 16)
datetime.datetime(2010, 11, 8, 0, 1, 16)
datetime.datetime(2010, 11, 9, 0, 1, 16)
datetime.datetime(2010, 11, 10, 0, 1, 16)
datetime.datetime(2010, 11, 11, 0, 5, 16)
datetime.datetime(2010, 11, 12, 0, 5, 16)
datetime.datetime(2010, 11, 13, 0, 5, 16)
datetime.datetime(2010, 11, 14, 0, 1, 16)
datetime.datetime(2010, 11, 15, 0, 1, 16)
datetime.datetime(2010, 11, 16, 0, 1, 16)
datetime.datetime(2010, 11, 17, 0, 5, 16)
datetime.datetime(2010, 11, 18, 0, 5, 16)
datetime.datetime(2010, 11, 19, 0, 5, 16)
datetime.datetime(2010, 11, 20, 0, 5, 16)
datetime.datetime(2010, 11, 21, 0, 5, 16)
datetime.datetime(2010, 11, 22, 0, 5, 16)
datetime.datetime(2010, 11, 23, 0, 5, 16)
datetime.datetime(2010, 11, 24, 0, 5, 16)
datetime.datetime(2010, 11, 25, 0, 5, 16)
datetime.datetime(2010, 11, 26, 0, 5, 16)
datetime.datetime(2010, 11, 27, 0, 1, 16)
```

```
 datetime.datetime(2010, 11, 28, 0, 1, 16)
 datetime.datetime(2010, 11, 29, 0, 1, 16)
 datetime.datetime(2010, 11, 30, 0, 1, 16)
 datetime.datetime(2010, 12, 1, 0, 1, 16)
 datetime.datetime(2010, 12, 2, 0, 1, 16)
 datetime.datetime(2010, 12, 3, 0, 1, 16)
 datetime.datetime(2010, 12, 4, 0, 1, 16)
 datetime.datetime(2010, 12, 5, 0, 1, 16)
 datetime.datetime(2010, 12, 6, 0, 1, 16)
 datetime.datetime(2010, 12, 7, 0, 1, 16)
 datetime.datetime(2010, 12, 8, 0, 1, 16)
 datetime.datetime(2010, 12, 9, 0, 1, 16)
 datetime.datetime(2010, 12, 10, 0, 1, 16)
 datetime.datetime(2010, 12, 11, 0, 1, 16)
 datetime.datetime(2010, 12, 12, 0, 1, 16)
 datetime.datetime(2010, 12, 13, 0, 1, 16)
 datetime.datetime(2010, 12, 14, 0, 1, 16)
 datetime.datetime(2010, 12, 15, 0, 1, 16)
 datetime.datetime(2010, 12, 16, 0, 1, 16)
 datetime.datetime(2010, 12, 17, 0, 5, 16)
 datetime.datetime(2010, 12, 18, 0, 5, 16)
 datetime.datetime(2010, 12, 19, 0, 5, 16)
 datetime.datetime(2010, 12, 20, 0, 5, 16)
 datetime.datetime(2010, 12, 21, 0, 1, 16)
 datetime.datetime(2010, 12, 22, 0, 1, 16)
 datetime.datetime(2010, 12, 23, 0, 1, 16)
 datetime.datetime(2010, 12, 24, 0, 1, 16)
 datetime.datetime(2010, 12, 25, 0, 1, 16)
 datetime.datetime(2010, 12, 26, 0, 1, 16)
 datetime.datetime(2010, 12, 27, 0, 1, 16)
 datetime.datetime(2010, 12, 28, 0, 1, 16)
 datetime.datetime(2010, 12, 29, 0, 1, 16)
 datetime.datetime(2010, 12, 30, 0, 1, 16)
 datetime.datetime(2010, 12, 31, 0, 1, 16)]
[datetime.datetime(2020, 1, 1, 0, 0, 32)
 datetime.datetime(2020, 1, 2, 0, 0, 32)
 datetime.datetime(2020, 1, 3, 0, 0, 32)
 datetime.datetime(2020, 1, 4, 0, 0, 32)
 datetime.datetime(2020, 1, 5, 0, 0, 32)
 datetime.datetime(2020, 1, 6, 0, 0, 32)
 datetime.datetime(2020, 1, 7, 0, 0, 32)
 datetime.datetime(2020, 1, 8, 0, 0, 32)
 datetime.datetime(2020, 1, 9, 0, 0, 32)
 datetime.datetime(2020, 1, 10, 0, 0, 32)
 datetime.datetime(2020, 1, 11, 0, 0, 32)
 datetime.datetime(2020, 1, 12, 0, 0, 32)
 datetime.datetime(2020, 1, 13, 0, 0, 32)
 datetime.datetime(2020, 1, 14, 0, 0, 32)
 datetime.datetime(2020, 1, 15, 0, 0, 32)
 datetime.datetime(2020, 1, 16, 0, 0, 32)
 datetime.datetime(2020, 1, 17, 0, 0, 32)
 datetime.datetime(2020, 1, 18, 0, 0, 32)
 datetime.datetime(2020, 1, 19, 0, 0, 32)
 datetime.datetime(2020, 1, 20, 0, 0, 32)
 datetime.datetime(2020, 1, 21, 0, 0, 32)
 datetime.datetime(2020, 1, 22, 0, 0, 32)
 datetime.datetime(2020, 1, 23, 0, 0, 32)
 datetime.datetime(2020, 1, 24, 0, 0, 32)
 datetime.datetime(2020, 1, 25, 0, 0, 32)
 datetime.datetime(2020, 1, 26, 0, 0, 32)
 datetime.datetime(2020, 1, 27, 0, 0, 32)
 datetime.datetime(2020, 1, 28, 0, 0, 32)
 datetime.datetime(2020, 1, 29, 0, 0, 32)
```

```
datetime.datetime(2020, 1, 30, 0, 0, 32)
datetime.datetime(2020, 1, 31, 0, 0, 32)
datetime.datetime(2020, 2, 1, 0, 0, 32)
datetime.datetime(2020, 2, 2, 0, 0, 32)
datetime.datetime(2020, 2, 3, 0, 0, 32)
datetime.datetime(2020, 2, 4, 0, 0, 32)
datetime.datetime(2020, 2, 5, 0, 0, 32)
datetime.datetime(2020, 2, 6, 0, 0, 32)
datetime.datetime(2020, 2, 7, 0, 0, 32)
datetime.datetime(2020, 2, 8, 0, 0, 32)
datetime.datetime(2020, 2, 9, 0, 0, 32)
datetime.datetime(2020, 2, 10, 0, 0, 32)
datetime.datetime(2020, 2, 11, 0, 0, 32)
datetime.datetime(2020, 2, 12, 0, 0, 32)
datetime.datetime(2020, 2, 13, 0, 0, 32)
datetime.datetime(2020, 2, 14, 0, 0, 32)
datetime.datetime(2020, 2, 15, 0, 0, 32)
datetime.datetime(2020, 2, 16, 0, 0, 32)
datetime.datetime(2020, 2, 17, 0, 0, 32)
datetime.datetime(2020, 2, 18, 0, 0, 32)
datetime.datetime(2020, 2, 19, 0, 0, 32)
datetime.datetime(2020, 2, 20, 0, 0, 32)
datetime.datetime(2020, 2, 21, 0, 0, 32)
datetime.datetime(2020, 2, 22, 0, 0, 32)
datetime.datetime(2020, 2, 23, 0, 0, 32)
datetime.datetime(2020, 2, 24, 0, 0, 32)
datetime.datetime(2020, 2, 25, 0, 0, 32)
datetime.datetime(2020, 2, 26, 0, 0, 32)
datetime.datetime(2020, 2, 27, 0, 0, 32)
datetime.datetime(2020, 2, 28, 0, 0, 32)
datetime.datetime(2020, 2, 29, 0, 0, 32)
datetime.datetime(2020, 3, 1, 0, 0, 32)
datetime.datetime(2020, 3, 2, 0, 0, 32)
datetime.datetime(2020, 3, 3, 0, 0, 32)
datetime.datetime(2020, 3, 4, 0, 0, 32)
datetime.datetime(2020, 3, 5, 0, 0, 32)
datetime.datetime(2020, 3, 6, 0, 0, 32)
datetime.datetime(2020, 3, 7, 0, 0, 32)
datetime.datetime(2020, 3, 8, 0, 0, 32)
datetime.datetime(2020, 3, 9, 0, 0, 32)
datetime.datetime(2020, 3, 10, 0, 0, 32)
datetime.datetime(2020, 3, 11, 0, 0, 32)
datetime.datetime(2020, 3, 12, 0, 0, 32)
datetime.datetime(2020, 3, 13, 0, 0, 32)
datetime.datetime(2020, 3, 14, 0, 0, 32)
datetime.datetime(2020, 3, 15, 0, 0, 32)
datetime.datetime(2020, 3, 16, 0, 0, 32)
datetime.datetime(2020, 3, 17, 0, 0, 32)
datetime.datetime(2020, 3, 18, 0, 0, 32)
datetime.datetime(2020, 3, 19, 0, 0, 32)
datetime.datetime(2020, 3, 20, 0, 0, 32)
datetime.datetime(2020, 3, 21, 0, 0, 32)
datetime.datetime(2020, 3, 22, 0, 0, 32)
datetime.datetime(2020, 3, 23, 0, 0, 32)
datetime.datetime(2020, 3, 24, 0, 0, 32)
datetime.datetime(2020, 3, 25, 0, 0, 32)
datetime.datetime(2020, 3, 26, 0, 0, 32)
datetime.datetime(2020, 3, 27, 0, 0, 32)
datetime.datetime(2020, 3, 28, 0, 0, 32)
datetime.datetime(2020, 3, 29, 0, 0, 32)
datetime.datetime(2020, 3, 30, 0, 0, 32)
datetime.datetime(2020, 3, 31, 0, 0, 32)
datetime.datetime(2020, 4, 1, 0, 0, 32)
```

```
datetime.datetime(2020, 4, 2, 0, 0, 32)
datetime.datetime(2020, 4, 3, 0, 0, 32)
datetime.datetime(2020, 4, 4, 0, 0, 32)
datetime.datetime(2020, 4, 5, 0, 0, 32)
datetime.datetime(2020, 4, 6, 0, 0, 32)
datetime.datetime(2020, 4, 7, 0, 0, 32)
datetime.datetime(2020, 4, 8, 0, 0, 32)
datetime.datetime(2020, 4, 9, 0, 0, 32)
datetime.datetime(2020, 4, 10, 0, 0, 32)
datetime.datetime(2020, 4, 11, 0, 0, 32)
datetime.datetime(2020, 4, 12, 0, 0, 32)
datetime.datetime(2020, 4, 13, 0, 0, 32)
datetime.datetime(2020, 4, 14, 0, 0, 32)
datetime.datetime(2020, 4, 15, 0, 0, 32)
datetime.datetime(2020, 4, 16, 0, 0, 32)
datetime.datetime(2020, 4, 17, 0, 0, 32)
datetime.datetime(2020, 4, 18, 0, 0, 32)
datetime.datetime(2020, 4, 19, 0, 0, 32)
datetime.datetime(2020, 4, 20, 0, 0, 32)
datetime.datetime(2020, 4, 21, 0, 0, 32)
datetime.datetime(2020, 4, 22, 0, 0, 54)
datetime.datetime(2020, 4, 23, 0, 0, 54)
datetime.datetime(2020, 4, 24, 0, 0, 54)
datetime.datetime(2020, 4, 25, 0, 0, 54)
datetime.datetime(2020, 4, 26, 0, 0, 54)
datetime.datetime(2020, 4, 27, 0, 0, 54)
datetime.datetime(2020, 4, 28, 0, 0, 54)
datetime.datetime(2020, 4, 29, 0, 0, 54)
datetime.datetime(2020, 4, 30, 0, 0, 54)
datetime.datetime(2020, 5, 1, 0, 0, 54)
datetime.datetime(2020, 5, 2, 0, 0, 54)
datetime.datetime(2020, 5, 3, 0, 0, 54)
datetime.datetime(2020, 5, 4, 0, 0, 54)
datetime.datetime(2020, 5, 5, 0, 0, 54)
datetime.datetime(2020, 5, 6, 0, 0, 54)
datetime.datetime(2020, 5, 7, 0, 0, 54)
datetime.datetime(2020, 5, 8, 0, 0, 54)
datetime.datetime(2020, 5, 9, 0, 0, 54)
datetime.datetime(2020, 5, 10, 0, 0, 54)
datetime.datetime(2020, 5, 11, 0, 0, 54)
datetime.datetime(2020, 5, 12, 0, 0, 54)
datetime.datetime(2020, 5, 13, 0, 0, 54)
datetime.datetime(2020, 5, 14, 0, 0, 54)
datetime.datetime(2020, 5, 15, 0, 0, 54)
datetime.datetime(2020, 5, 16, 0, 0, 54)
datetime.datetime(2020, 5, 17, 19, 55)
datetime.datetime(2020, 5, 18, 0, 1, 13)
datetime.datetime(2020, 5, 19, 0, 1, 13)
datetime.datetime(2020, 5, 20, 0, 1, 13)
datetime.datetime(2020, 5, 21, 0, 1, 13)
datetime.datetime(2020, 5, 22, 0, 1, 13)
datetime.datetime(2020, 5, 23, 0, 1, 13)
datetime.datetime(2020, 5, 24, 0, 1, 13)
datetime.datetime(2020, 5, 25, 0, 1, 13)
datetime.datetime(2020, 5, 26, 0, 1, 13)
datetime.datetime(2020, 5, 27, 0, 1, 13)
datetime.datetime(2020, 5, 28, 0, 1, 13)
datetime.datetime(2020, 5, 29, 0, 1, 13)
datetime.datetime(2020, 5, 30, 0, 1, 13)
datetime.datetime(2020, 5, 31, 0, 1, 13)
datetime.datetime(2020, 6, 1, 0, 1, 13)
datetime.datetime(2020, 6, 2, 0, 1, 13)
datetime.datetime(2020, 6, 3, 0, 1, 13)
```

```
datetime.datetime(2020, 6, 4, 0, 1, 13)
datetime.datetime(2020, 6, 5, 0, 1, 13)
datetime.datetime(2020, 6, 6, 0, 1, 13)
datetime.datetime(2020, 6, 7, 0, 1, 13)
datetime.datetime(2020, 6, 8, 0, 1, 13)
datetime.datetime(2020, 6, 9, 0, 1, 13)
datetime.datetime(2020, 6, 10, 0, 1, 13)
datetime.datetime(2020, 6, 11, 1, 37, 13)
datetime.datetime(2020, 6, 12, 0, 1, 13)
datetime.datetime(2020, 6, 13, 0, 1, 13)
datetime.datetime(2020, 6, 14, 0, 1, 13)
datetime.datetime(2020, 6, 15, 0, 1, 13)
datetime.datetime(2020, 6, 16, 0, 1, 13)
datetime.datetime(2020, 6, 17, 0, 1, 13)
datetime.datetime(2020, 6, 18, 0, 1, 13)
datetime.datetime(2020, 6, 19, 0, 1, 13)
datetime.datetime(2020, 6, 20, 0, 1, 13)
datetime.datetime(2020, 6, 21, 0, 1, 13)
datetime.datetime(2020, 6, 22, 0, 1, 13)
datetime.datetime(2020, 6, 23, 0, 1, 13)
datetime.datetime(2020, 6, 24, 0, 3, 10)
datetime.datetime(2020, 6, 25, 0, 1, 35)
datetime.datetime(2020, 6, 26, 0, 1, 35)
datetime.datetime(2020, 6, 27, 0, 1, 35)
datetime.datetime(2020, 6, 28, 0, 1, 35)
datetime.datetime(2020, 6, 29, 0, 1, 35)
datetime.datetime(2020, 6, 30, 0, 1, 35)
datetime.datetime(2020, 7, 1, 0, 1, 35)
datetime.datetime(2020, 7, 2, 0, 1, 35)
datetime.datetime(2020, 7, 3, 0, 1, 35)
datetime.datetime(2020, 7, 4, 0, 1, 35)
datetime.datetime(2020, 7, 5, 0, 1, 35)
datetime.datetime(2020, 7, 6, 0, 1, 35)
datetime.datetime(2020, 7, 7, 0, 1, 35)
datetime.datetime(2020, 7, 8, 0, 1, 35)
datetime.datetime(2020, 7, 9, 0, 1, 35)
datetime.datetime(2020, 7, 10, 0, 1, 35)
datetime.datetime(2020, 7, 11, 0, 1, 35)
datetime.datetime(2020, 7, 12, 0, 1, 35)
datetime.datetime(2020, 7, 13, 0, 1, 35)
datetime.datetime(2020, 7, 14, 0, 1, 35)
datetime.datetime(2020, 7, 15, 0, 1, 35)
datetime.datetime(2020, 7, 16, 0, 1, 35)
datetime.datetime(2020, 7, 17, 0, 1, 35)
datetime.datetime(2020, 7, 18, 0, 1, 35)
datetime.datetime(2020, 7, 19, 0, 1, 35)
datetime.datetime(2020, 7, 20, 0, 1, 35)
datetime.datetime(2020, 7, 21, 0, 1, 35)
datetime.datetime(2020, 7, 22, 0, 1, 35)
datetime.datetime(2020, 7, 23, 0, 1, 35)
datetime.datetime(2020, 7, 24, 0, 1, 35)
datetime.datetime(2020, 7, 25, 0, 1, 35)
datetime.datetime(2020, 7, 26, 0, 1, 35)
datetime.datetime(2020, 7, 27, 0, 1, 35)
datetime.datetime(2020, 7, 28, 0, 1, 35)
datetime.datetime(2020, 7, 29, 0, 1, 35)
datetime.datetime(2020, 7, 30, 0, 1, 35)
datetime.datetime(2020, 7, 31, 0, 1, 35)
datetime.datetime(2020, 8, 1, 0, 1, 35)
datetime.datetime(2020, 8, 2, 0, 1, 35)
datetime.datetime(2020, 8, 3, 0, 1, 35)
datetime.datetime(2020, 8, 4, 0, 1, 35)
datetime.datetime(2020, 8, 5, 0, 1, 35)
```

```
datetime.datetime(2020, 8, 6, 0, 1, 35)
datetime.datetime(2020, 8, 7, 0, 1, 35)
datetime.datetime(2020, 8, 8, 0, 1, 35)
datetime.datetime(2020, 8, 9, 0, 1, 35)
datetime.datetime(2020, 8, 10, 0, 1, 35)
datetime.datetime(2020, 8, 11, 0, 1, 35)
datetime.datetime(2020, 8, 12, 0, 1, 35)
datetime.datetime(2020, 8, 13, 0, 1, 35)
datetime.datetime(2020, 8, 14, 0, 1, 35)
datetime.datetime(2020, 8, 15, 0, 1, 35)
datetime.datetime(2020, 8, 16, 0, 1, 35)
datetime.datetime(2020, 8, 17, 0, 1, 35)
datetime.datetime(2020, 8, 18, 0, 1, 35)
datetime.datetime(2020, 8, 19, 0, 1, 35)
datetime.datetime(2020, 8, 20, 0, 1, 35)
datetime.datetime(2020, 8, 21, 0, 1, 35)
datetime.datetime(2020, 8, 22, 0, 1, 35)
datetime.datetime(2020, 8, 23, 0, 1, 35)
datetime.datetime(2020, 8, 24, 0, 1, 35)
datetime.datetime(2020, 8, 25, 0, 1, 35)
datetime.datetime(2020, 8, 26, 0, 1, 35)
datetime.datetime(2020, 8, 27, 0, 1, 35)
datetime.datetime(2020, 8, 28, 0, 1, 35)
datetime.datetime(2020, 8, 29, 0, 1, 35)
datetime.datetime(2020, 8, 30, 0, 1, 35)
datetime.datetime(2020, 8, 31, 0, 1, 35)
datetime.datetime(2020, 9, 1, 0, 1, 35)
datetime.datetime(2020, 9, 2, 0, 1, 35)
datetime.datetime(2020, 9, 3, 0, 1, 35)
datetime.datetime(2020, 9, 4, 0, 1, 35)
datetime.datetime(2020, 9, 5, 0, 1, 35)
datetime.datetime(2020, 9, 6, 0, 1, 35)
datetime.datetime(2020, 9, 7, 0, 1, 35)
datetime.datetime(2020, 9, 8, 0, 1, 35)
datetime.datetime(2020, 9, 9, 0, 1, 35)
datetime.datetime(2020, 9, 10, 0, 1, 35)
datetime.datetime(2020, 9, 11, 0, 1, 35)
datetime.datetime(2020, 9, 12, 0, 1, 35)
datetime.datetime(2020, 9, 13, 0, 1, 35)
datetime.datetime(2020, 9, 14, 0, 1, 35)
datetime.datetime(2020, 9, 15, 0, 1, 35)
datetime.datetime(2020, 9, 16, 0, 1, 35)
datetime.datetime(2020, 9, 17, 0, 1, 35)
datetime.datetime(2020, 9, 18, 0, 1, 35)
datetime.datetime(2020, 9, 19, 0, 1, 35)
datetime.datetime(2020, 9, 20, 0, 1, 35)
datetime.datetime(2020, 9, 21, 0, 1, 35)
datetime.datetime(2020, 9, 22, 4, 29, 35)
datetime.datetime(2020, 9, 23, 0, 1, 35)
datetime.datetime(2020, 9, 24, 0, 1, 35)
datetime.datetime(2020, 9, 25, 0, 1, 35)
datetime.datetime(2020, 9, 26, 0, 1, 35)
datetime.datetime(2020, 9, 27, 0, 1, 35)
datetime.datetime(2020, 9, 28, 0, 1, 35)
datetime.datetime(2020, 9, 29, 0, 1, 35)
datetime.datetime(2020, 9, 30, 0, 1, 35)
datetime.datetime(2020, 10, 1, 0, 1, 35)
datetime.datetime(2020, 10, 2, 0, 1, 35)
datetime.datetime(2020, 10, 3, 0, 1, 35)
datetime.datetime(2020, 10, 4, 0, 1, 35)
datetime.datetime(2020, 10, 5, 0, 1, 35)
datetime.datetime(2020, 10, 6, 0, 1, 35)
datetime.datetime(2020, 10, 7, 0, 1, 35)
```

```
datetime.datetime(2020, 10, 8, 0, 1, 35)
datetime.datetime(2020, 10, 9, 0, 1, 35)
datetime.datetime(2020, 10, 10, 0, 1, 35)
datetime.datetime(2020, 10, 11, 0, 1, 35)
datetime.datetime(2020, 10, 12, 0, 1, 35)
datetime.datetime(2020, 10, 13, 0, 1, 35)
datetime.datetime(2020, 10, 14, 0, 1, 35)
datetime.datetime(2020, 10, 15, 0, 1, 35)
datetime.datetime(2020, 10, 16, 0, 1, 35)
datetime.datetime(2020, 10, 17, 0, 1, 35)
datetime.datetime(2020, 10, 18, 0, 1, 35)
datetime.datetime(2020, 10, 19, 0, 1, 35)
datetime.datetime(2020, 10, 20, 0, 1, 35)
datetime.datetime(2020, 10, 21, 0, 1, 35)
datetime.datetime(2020, 10, 22, 0, 1, 35)
datetime.datetime(2020, 10, 23, 0, 1, 35)
datetime.datetime(2020, 10, 24, 0, 1, 35)
datetime.datetime(2020, 10, 25, 0, 1, 35)
datetime.datetime(2020, 10, 26, 0, 1, 35)
datetime.datetime(2020, 10, 27, 0, 1, 35)
datetime.datetime(2020, 10, 28, 0, 1, 35)
datetime.datetime(2020, 10, 29, 0, 1, 35)
datetime.datetime(2020, 10, 30, 0, 1, 35)
datetime.datetime(2020, 10, 31, 0, 1, 35)
datetime.datetime(2020, 11, 1, 0, 1, 35)
datetime.datetime(2020, 11, 2, 0, 1, 35)
datetime.datetime(2020, 11, 3, 0, 1, 35)
datetime.datetime(2020, 11, 4, 0, 1, 35)
datetime.datetime(2020, 11, 5, 0, 1, 35)
datetime.datetime(2020, 11, 6, 0, 1, 35)
datetime.datetime(2020, 11, 7, 0, 1, 35)
datetime.datetime(2020, 11, 8, 0, 1, 35)
datetime.datetime(2020, 11, 9, 0, 1, 35)
datetime.datetime(2020, 11, 10, 0, 1, 35)
datetime.datetime(2020, 11, 11, 0, 1, 35)
datetime.datetime(2020, 11, 12, 0, 1, 35)
datetime.datetime(2020, 11, 13, 0, 1, 35)
datetime.datetime(2020, 11, 14, 0, 1, 35)
datetime.datetime(2020, 11, 15, 0, 1, 35)
datetime.datetime(2020, 11, 16, 0, 1, 35)
datetime.datetime(2020, 11, 17, 0, 1, 35)
datetime.datetime(2020, 11, 18, 0, 1, 35)
datetime.datetime(2020, 11, 19, 0, 1, 35)
datetime.datetime(2020, 11, 20, 0, 1, 35)
datetime.datetime(2020, 11, 21, 0, 1, 35)
datetime.datetime(2020, 11, 22, 0, 1, 35)
datetime.datetime(2020, 11, 23, 0, 1, 35)
datetime.datetime(2020, 11, 24, 0, 1, 35)
datetime.datetime(2020, 11, 25, 0, 1, 35)
datetime.datetime(2020, 11, 26, 0, 1, 35)
datetime.datetime(2020, 11, 27, 0, 1, 35)
datetime.datetime(2020, 11, 28, 0, 1, 35)
datetime.datetime(2020, 11, 29, 0, 1, 35)
datetime.datetime(2020, 11, 30, 0, 1, 35)
datetime.datetime(2020, 12, 1, 0, 1, 35)
datetime.datetime(2020, 12, 2, 0, 1, 35)
datetime.datetime(2020, 12, 3, 0, 1, 35)
datetime.datetime(2020, 12, 4, 0, 1, 35)
datetime.datetime(2020, 12, 5, 0, 1, 35)
datetime.datetime(2020, 12, 6, 0, 1, 35)
datetime.datetime(2020, 12, 7, 0, 1, 35)
datetime.datetime(2020, 12, 8, 0, 1, 35)
datetime.datetime(2020, 12, 9, 0, 1, 35)
```

```
 datetime.datetime(2020, 12, 10, 0, 1, 35)
 datetime.datetime(2020, 12, 11, 0, 1, 35)
 datetime.datetime(2020, 12, 12, 0, 1, 35)
 datetime.datetime(2020, 12, 13, 0, 1, 35)
 datetime.datetime(2020, 12, 14, 0, 1, 35)
 datetime.datetime(2020, 12, 15, 0, 1, 35)
 datetime.datetime(2020, 12, 16, 0, 1, 35)
 datetime.datetime(2020, 12, 17, 0, 1, 35)
 datetime.datetime(2020, 12, 18, 0, 1, 35)
 datetime.datetime(2020, 12, 19, 0, 1, 35)
 datetime.datetime(2020, 12, 20, 0, 1, 35)
 datetime.datetime(2020, 12, 21, 0, 1, 35)
 datetime.datetime(2020, 12, 22, 0, 1, 35)
 datetime.datetime(2020, 12, 23, 0, 1, 35)
 datetime.datetime(2020, 12, 24, 0, 1, 35)
 datetime.datetime(2020, 12, 25, 0, 1, 35)
 datetime.datetime(2020, 12, 26, 0, 1, 35)
 datetime.datetime(2020, 12, 27, 0, 1, 35)
 datetime.datetime(2020, 12, 28, 0, 1, 35)
 datetime.datetime(2020, 12, 29, 0, 1, 35)
 datetime.datetime(2020, 12, 30, 0, 1, 35)
 datetime.datetime(2020, 12, 31, 0, 1, 35)]
Number of subsampled observations in 2010: 349
Number of subsampled observations in 2020: 366
```

In [19]:

```python
print(f'Subsampled 2010 mean: {sub10mean: .4f} +/- {sub10err: .4f} deg C\
print(f'Subsampled 2020 mean: {sub20mean: .4f} +/- {sub20err: .4f} deg C

y = 0


if mean10+err10 >= sub10mean and mean10-err10 <= sub10mean:
    # this means that full mean falls inside subsample error bars
    print('2010 Subsampling within Full Dataset Error Bars')
    y = y + 1
else:
    print('2010 Subsampling outside of Full Dataset Error Bars')


if mean20+err20 >= sub20mean and mean20-err20 <= sub20mean:
    # this means that full mean falls inside subsample error bars
    print('2020 Subsampling within Full Dataset Error Bars')
    y = y + 1
else:
    print('2020 Subsampling outside of Full Dataset Error Bars')

if y > 2:
    print('Subsample means are consistent')
elif y < 2:
    print('Subsample means are not consistent')
```

```
Subsampled 2010 mean:  16.8968 +/-  0.0749 deg C
 versus full dataset 2010 mean of  16.5155 +/-  0.0078 deg C
Subsampled 2020 mean:  18.3642 +/-  0.0845 deg C
 versus full dataset 2020 mean of  17.7528 +/-  0.0071 deg C
2010 Subsampling outside of Full Dataset Error Bars
2020 Subsampling outside of Full Dataset Error Bars
Subsample means are not consistent
```

Yes - the subsampled means are different. They fall outside of the error bars for the full dataset mean. This makes sense because we have much less data for the subsampling, and therefore our mean is more uncertain (larger standard error).

Q3: Variance

For each quantity, evaluate whether the variance from the two years are consistent. Do you obtain different results if you subsample the data at (say) once per day?

In [21]:

```
# found variance previously for full dataset (see above)


sum_sub10 = 0; S10 = 0; # sum10 = sig^2 and N is number of observations
for val in sub10:
    sig = abs(val-sub10mean)
    sig2 = np.power(sig,2)
    sum_sub10 = sum_sub10 + sig2
    S10 = S10 + 1

sum_sub20 = 0; S20 = 0;
for val in sub20:
    sig = abs(val-sub20mean)
    sig2 = np.power(sig,2)
    sum_sub20 = sum_sub20 + sig2
    S20 = S20 + 1


var_sub10 = sum_sub10/S10; sig_sub10 = np.sqrt(var_sub10); err_sub10 = s
var_sub20 = sum_sub20/S20; sig_sub20 = np.sqrt(var_sub20); err_sub20 = s


print(f'2010 variance = {var10: .3f} and std error from mean = {err10: .
print(f'2020 variance = {var20: .3f} and std error from mean = {err20: .


print(f'Subsampled 2010 variance = {var_sub10: .3f} and std error from m
print(f'Subsampled 2020 variance = {var_sub20: .3f} and std error from m
```

```
2010 variance =  3.726 and std error from mean =  0.008
2020 variance =  6.512 and std error from mean =  0.007
Subsampled 2010 variance =  3.840 and std error from mean =  0.105
Subsampled 2020 variance =  6.845 and std error from mean =  0.137
```

The variance and standard error of the subsampled data is greater than the full dataset. The variance is more simular, because the variability in the temperature measurements is roughly similar. On the other hand, the standard error from the mean is much bigger for the subsampled dataset because there is much less data to correctly constrain the mean, so it is more uncertain (larger error).

Q4: Extreme values

For each of the two years, compute the likelihood given the observed PDF, and a Gaussian PDF with the observed variance, of a temperature extreme value 3σ greater than the mean

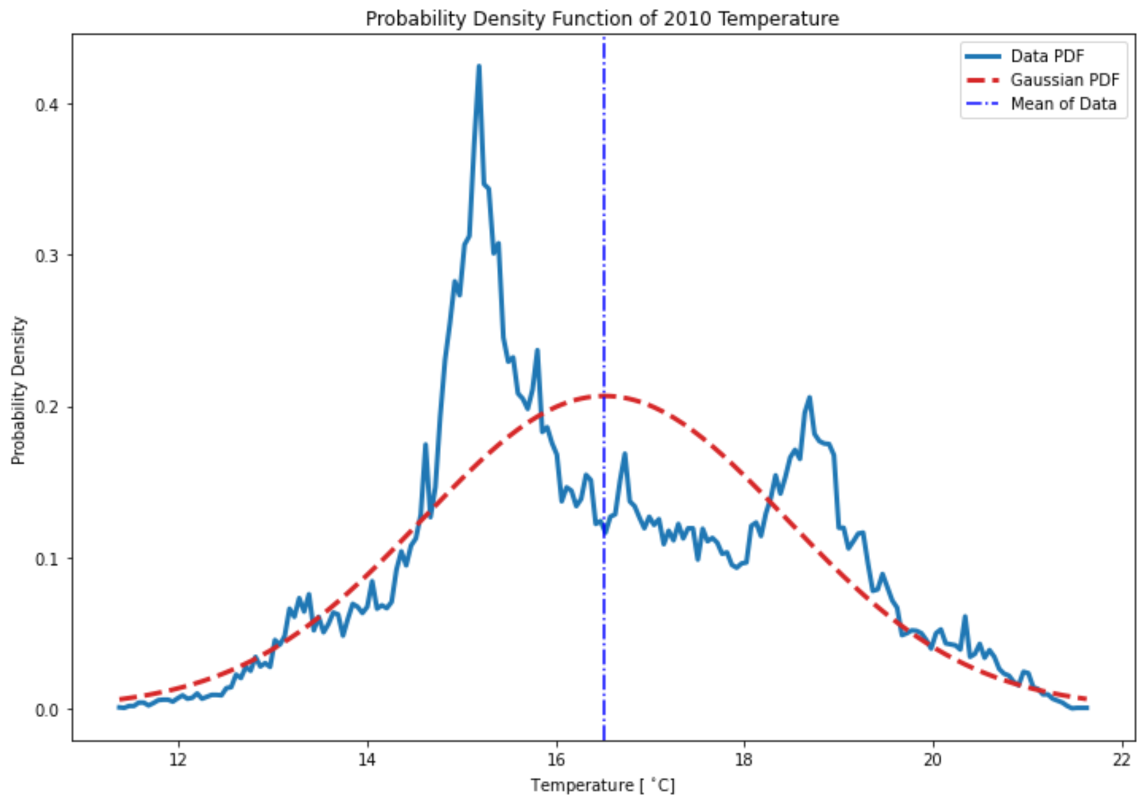
In [22]:

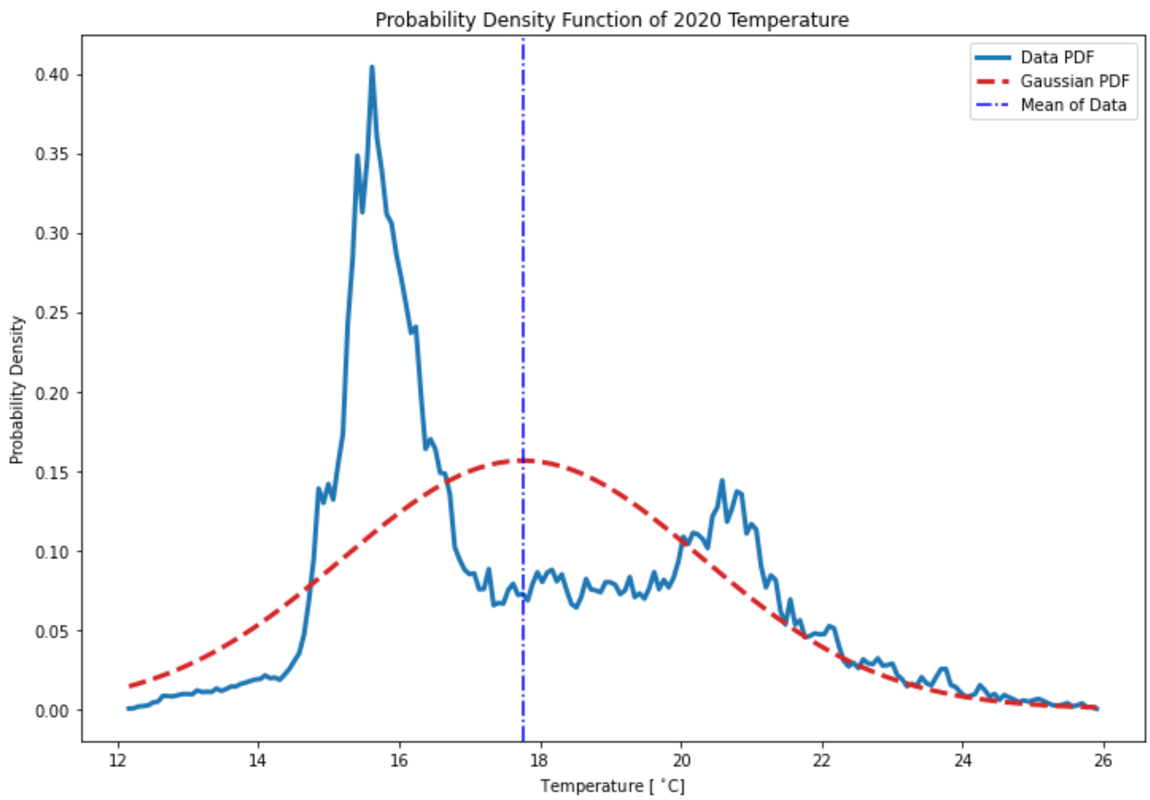
```
# let's start by finding what an extreme value (3*sigma) would look like

# for 2010 dataset:

extreme10 = 3*sig10 + mean10
extreme20 = 3*sig20 + mean20

print(f'Extreme value for 2010 Dataset is: {extreme10: .3f} deg C')
print(f'Extreme value for 2020 Dataset is: {extreme20: .3f} deg C')
```

```
Extreme value for 2010 Dataset is:  22.306 deg C
Extreme value for 2020 Dataset is:  25.408 deg C
```

In [23]:

```
# make a PDF, and then a Gaussian PDF

nbins = 200
[bin10,bin10_edges] = np.histogram(t10,bins=nbins)
[bin20,bin20_edges] = np.histogram(t20,bins=nbins)

# total number of observations
N10 = sum(bin10) # number of observations in 2010 dataset
N20 = sum(bin20) # number of observations in 2020 dataset

# distance between bins
d10 = bin10_edges[1:]-bin10_edges[0:-1]; d10 = np.nanmean(d10)
d20 = bin20_edges[1:]-bin20_edges[0:-1]; d20 = np.nanmean(d20)

# midpoint values of each bin (deg C)
mid10 = (bin10_edges[1:]+bin10_edges[0:-1])/2
mid20 = (bin20_edges[1:]+bin20_edges[0:-1])/2

pdf10 = bin10/N10/d10
pdf20 = bin20/N20/d20

gaus10 = (1/sig10/np.sqrt(2*np.pi)) * (np.exp(-(mid10-mean10)**2 / (2*si
gaus20 = (1/sig20/np.sqrt(2*np.pi)) * (np.exp(-(mid20-mean20)**2 / (2*si

year_strings = ['2010','2020']
x = [mid10,mid20]
y = [pdf10,pdf20]
yg = [gaus10,gaus20]
mu = [mean10,mean20]
fig,axes = plt.subplots(2,1,figsize=(12,18))
for n,ax in enumerate(axes):
    ax.set_title(f'Probability Density Function of {year_strings[n]} Tem
    ax.plot(x[n],y[n],color='tab:blue',label='Data PDF',linewidth=3)
    ax.plot(x[n],yg[n],color='tab:red',linestyle='--',label='Gaussian PD
    ax.set_xlabel(r'Temperature [ $^{\circ} $C]'); ax.set_ylabel('Probab
    ax.axvline(mu[n],color='blue',linestyle='-.',label='Mean of Data')
    ax.legend()

plt.show()

# check that all of the PDF integrals go to 1
# check: PDF must add to 1
print(f'Sum of 2010 PDF = {np.sum(pdf10)*d10}')
print(f'Sum of 2020 PDF = {np.sum(pdf20)*d20}')

print(f'Sum of 2010 Gaussian PDF = {np.sum(gaus10)*d10}')
print(f'Sum of 2020 Gaussian PDF = {np.sum(gaus20)*d20}')
```

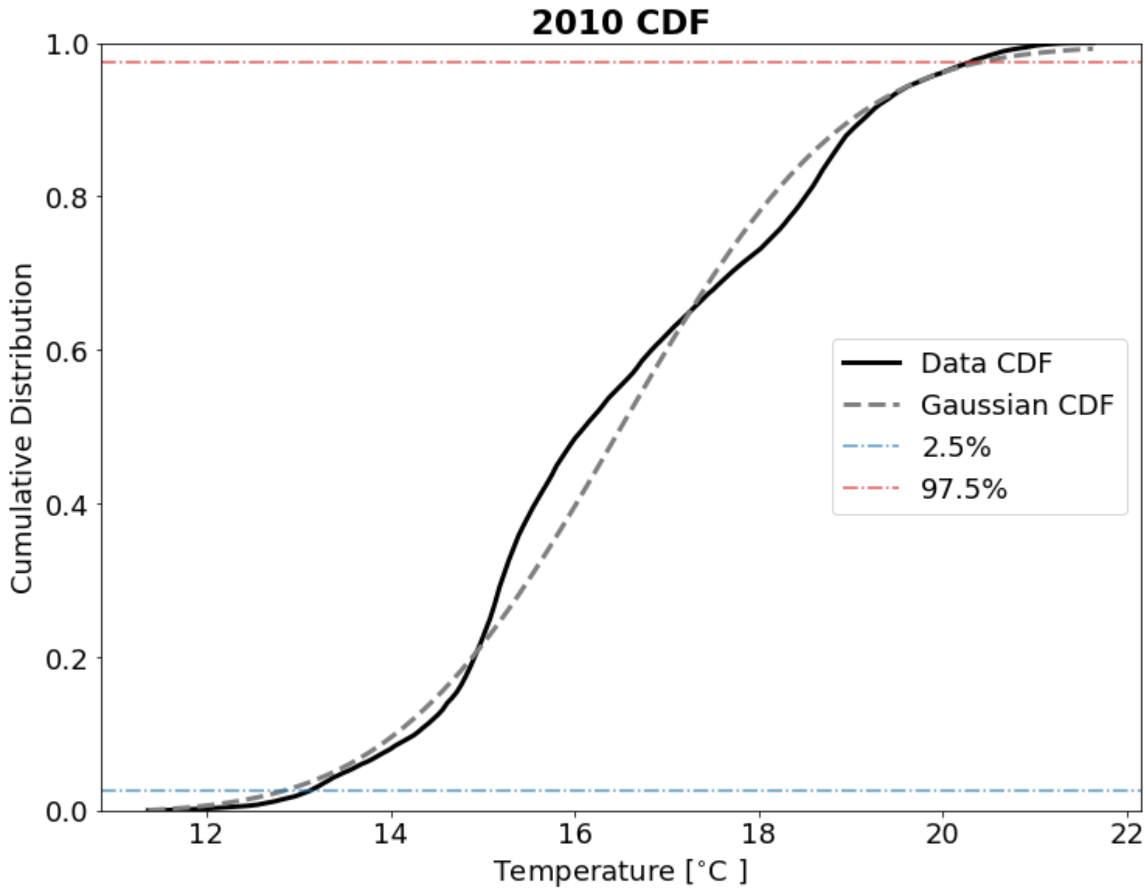


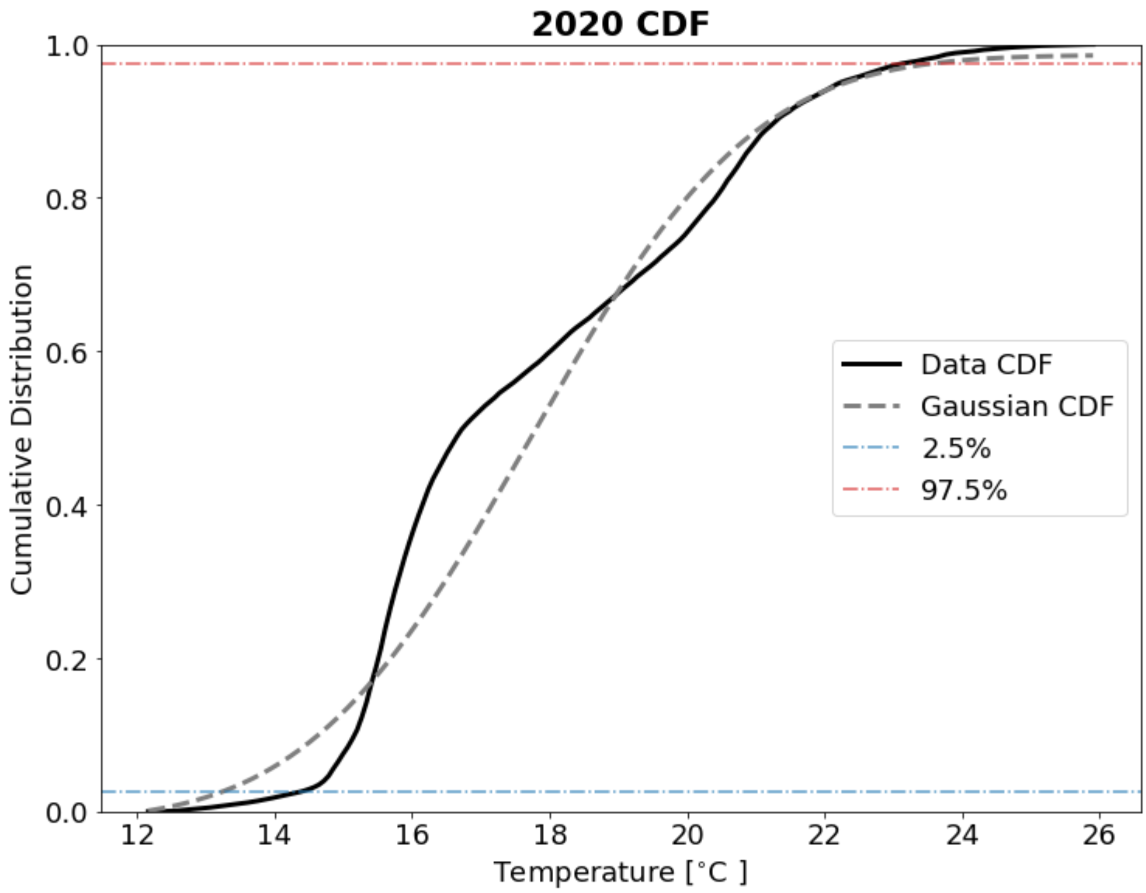



```
Sum of 2010 PDF = 0.9999999999999998
Sum of 2020 PDF = 1.0
Sum of 2010 Gaussian PDF = 0.992438018321991
Sum of 2020 Gaussian PDF = 0.9855733513832092
```

In [24]:

```
# calculating CDF: integral of PDF
# therefore: CDF = \int_a ^b PDF(T) dT = (b-a) * ((PDF(a)+PDF(b))/2)

# cumulative distribution function is the integral of the PDF
# so: cdf = dbin * (PDF(a)+PDF(b))/2
cdf10 = np.cumsum(pdf10)*d10
cdf20 = np.cumsum(pdf20)*d20

# gaussian CDF
gc10 = np.cumsum(gaus10)*d10
gc20 = np.cumsum(gaus20)*d20

plt.rcParams['font.size'] = 18

year_strings = ['2010','2020']
x = [mid10,mid20]
y = [cdf10,cdf20]
gy = [gc10,gc20]
fig,axes = plt.subplots(2,1,figsize=(12,20))
for n,ax in enumerate(axes):
    ax.set_title(f'{year_strings[n]} CDF',fontweight='bold')
    ax.plot(x[n],y[n],color='black',linewidth=3,label='Data CDF')
    ax.plot(x[n],gy[n],color='tab:gray',linewidth=3,linestyle='--',label
    ax.axhline(y=0.025,color='tab:blue',alpha=0.75,linestyle='-.',label=
    ax.axhline(y=0.975,color='tab:red',alpha=0.75,linestyle='-.',label='
    ax.set_ylim([0,1])
    ax.set_xlabel(r'Temperature [$^{\circ}$C ]')
    ax.set_ylabel('Cumulative Distribution')
    ax.legend(loc='best')

plt.show()
```


2010 CDF
1.0
0.8
0.6
0.4
0.2
0.0
Cumulative Distribution
12
14
16
18
20
22
Temperature [°C]
Data CDF
Gaussian CDF
2.5%
97.5%


2020 CDF
1.0
0.8
0.6
0.4
0.2
0.0
Cumulative Distribution
12
14
16
18
20
22
24
26
Temperature [°C]
Data CDF
Gaussian CDF
2.5%
97.5%

In [25]:

```
# evaluate CDF at 3*sigma

def ind_nearest(array, value):
    array = np.asarray(array)
    idx = (np.abs(array - value)).argmin()
    return idx

idx10 = ind_nearest(mid10,extreme10)
prob_ex10 = pdf10[idx10]
gaus_ex10 = gaus10[idx10]

idx20 = ind_nearest(mid20,extreme20)
prob_ex20 = pdf20[idx20]
gaus_ex20 = gaus20[idx20]


print('Real Data PDF:')
# print(prob_ex10)
print(f'Likelihood of finding 3*sigma greater than mean for 2010: {prob_

# print(prob_ex20)
print(f'Likelihood of finding 3*sigma greater than mean for 2020: {prob_

print('\n')

print('Gaussian PDF:')
# print(prob_ex10)
print(f'Likelihood of finding 3*sigma greater than mean for 2010: {gaus_

# print(prob_ex20)
print(f'Likelihood of finding 3*sigma greater than mean for 2020: {gaus_
```

```
Real Data PDF:
Likelihood of finding 3*sigma greater than mean for 2010:  0.03185368 %
Likelihood of finding 3*sigma greater than mean for 2020:  0.26649552 %


Gaussian PDF:
Likelihood of finding 3*sigma greater than mean for 2010:  0.61887698 %
Likelihood of finding 3*sigma greater than mean for 2020:  0.17165564 %
```

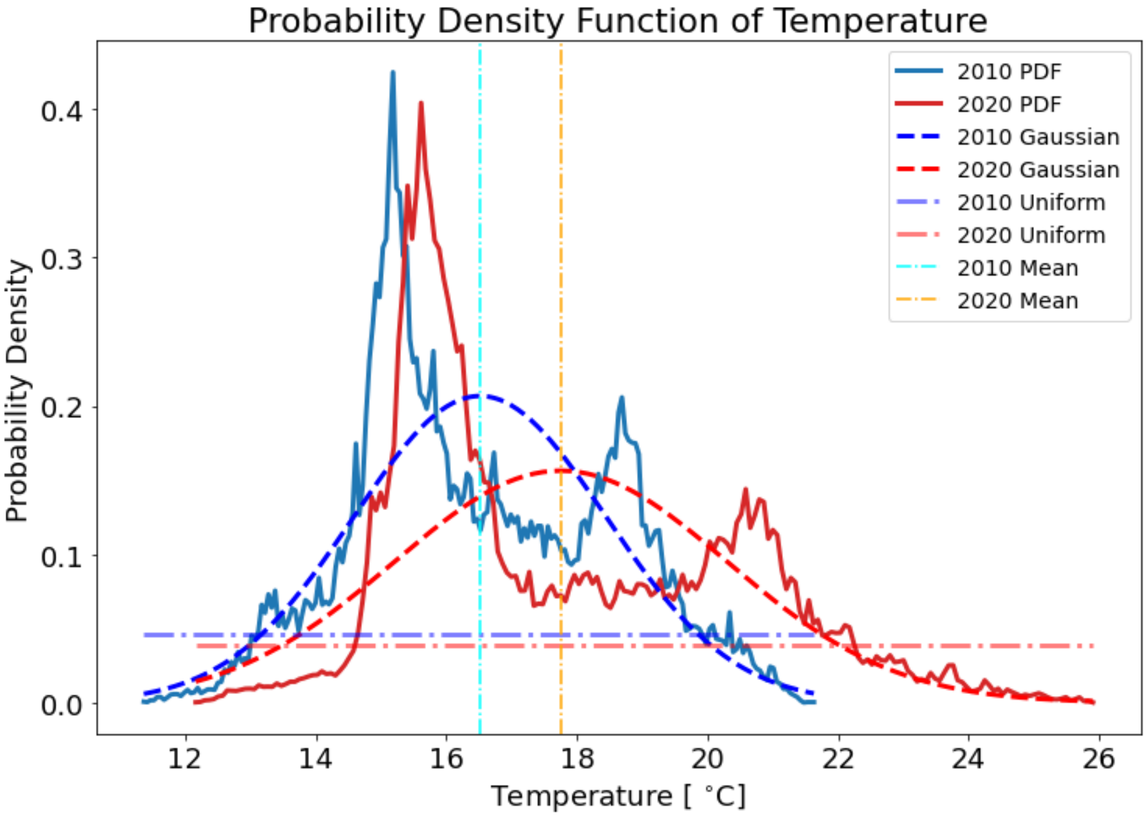
Q5: Probability Density Functions

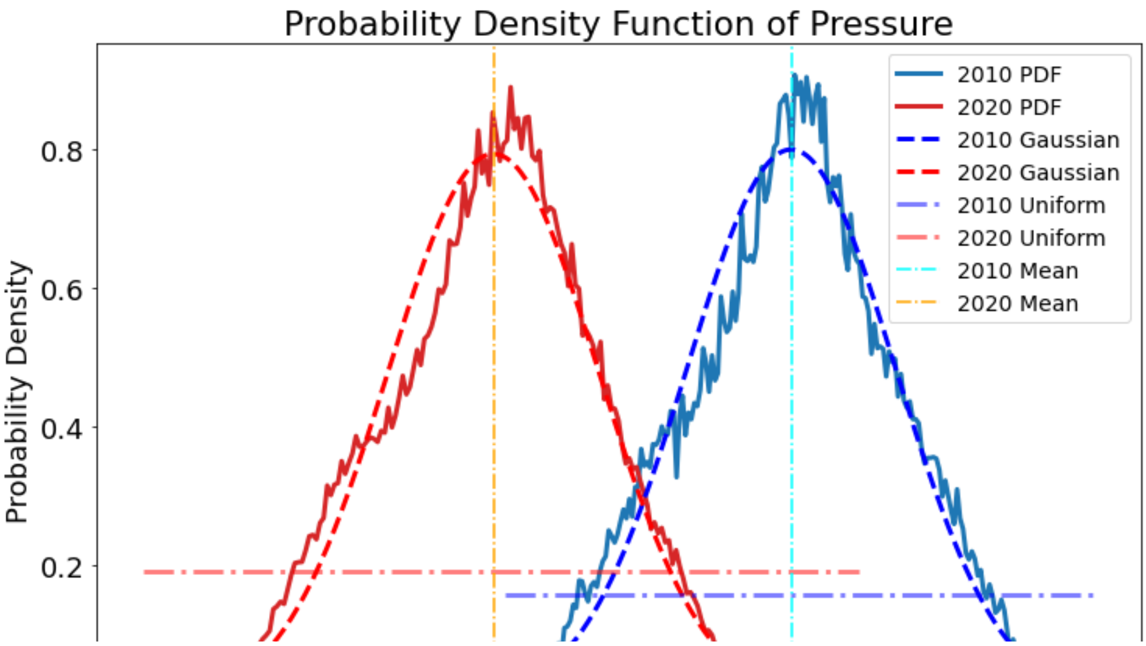
For both pressure and temperature, plot the pdfs for the two years on the same axes. Overplot the Gaussian and uniform distributions that have the observed mean and variance.

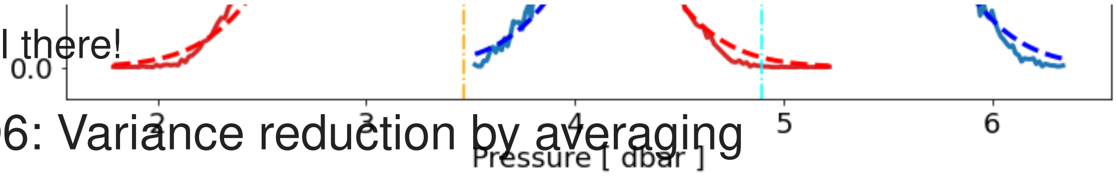
In [26]:

```
# make a PDF, and then a Gaussian PDF, and then uniform distribution PDF

def data_PDF(data,nbins):
    import numpy as np
    [num_per_bin, bin_edges] = np.histogram(data,bins=nbins)
    N = sum(num_per_bin)
    dbin = bin_edges[1:]-bin_edges[0:-1]; dbin = np.nanmean(dbin)
    mid_bins = (bin_edges[0:-1]+bin_edges[1:])/2
    pdf = num_per_bin/N/dbin
    return N, mid_bins, pdf

def get_stats(data):
    import numpy as np
    m = np.nanmean(data)
    stdev = np.std(data)
    return m,stdev

nbins = 200
[N_t10,mid_t10,pdf_t10] = data_PDF(t10,nbins)
[N_t20,mid_t20,pdf_t20] = data_PDF(t20,nbins)
[N_p10,mid_p10,pdf_p10] = data_PDF(p10,nbins)
[N_p20,mid_p20,pdf_p20] = data_PDF(p20,nbins)


[meant10,stdt10] = get_stats(t10)
[meant20,stdt20] = get_stats(t20)
[meanp10,stdp10] = get_stats(p10)
[meanp20,stdp20] = get_stats(p20)


# now just using scipy functions

# Gaussian
# syntax: scipy.stats.norm.pdf(x)

gaust10 = stats.norm.pdf(mid_t10,loc=meant10,scale=stdt10)
gaust20 = stats.norm.pdf(mid_t20,loc=meant20,scale=stdt20)

gausp10 = stats.norm.pdf(mid_p10,loc=meanp10,scale=stdp10)
gausp20 = stats.norm.pdf(mid_p20,loc=meanp20,scale=stdp20)

# uniform distribution
# syntax: scipy.stats.uniform.pdf(x)

unit10 = stats.uniform.pdf(mid_t10,np.nanmin(t10),np.nanmax(t10))
unit20 = stats.uniform.pdf(mid_t20,np.nanmin(t20),np.nanmax(t20))

unip10 = stats.uniform.pdf(mid_p10,np.nanmin(p10),np.nanmax(p10))
unip20 = stats.uniform.pdf(mid_p20,np.nanmin(p20),np.nanmax(p20))


# plotting:

fig,(ax1,ax2) = plt.subplots(2,1,figsize=(12,18))
ax1.set_title(f'Probability Density Function of Temperature')
ax1.plot(mid_t10,pdf_t10,color='tab:blue',label='2010 PDF',linewidth=3)
ax1.plot(mid_t20,pdf_t20,color='tab:red',label='2020 PDF',linewidth=3)
ax1.plot(mid_t10,gaust10,color='blue',linestyle='--',label='2010 Gaussia
ax1.plot(mid_t20,gaust20,color='red',linestyle='--',label='2020 Gaussian
ax1.plot(mid_t10,unit10,color='blue',linestyle='-.',alpha=0.5,label='201
ax1.plot(mid_t20,unit20,color='red',linestyle='-.',alpha=0.5,label='2020
ax1.set_xlabel(r'Temperature [ $^{\circ} $C]'); ax1.set_ylabel('Probabil
```

```
ax1.axvline(x=meant10,color='cyan',linestyle='-.',label='2010 Mean')
ax1.axvline(x=meant20,color='orange',linestyle='-.',label='2020 Mean')
ax1.legend(loc='best',fontsize=14)

ax2.set_title(f'Probability Density Function of Pressure')
ax2.plot(mid_p10,pdf_p10,color='tab:blue',label='2010 PDF',linewidth=3)
ax2.plot(mid_p20,pdf_p20,color='tab:red',label='2020 PDF',linewidth=3)
ax2.plot(mid_p10,gausp10,color='blue',linestyle='--',label='2010 Gaussia
ax2.plot(mid_p20,gausp20,color='red',linestyle='--',label='2020 Gaussian
ax2.plot(mid_p10,unip10,color='blue',linestyle='-.',alpha=0.5,label='201
ax2.plot(mid_p20,unip20,color='red',linestyle='-.',alpha=0.5,label='2020
ax2.set_xlabel('Pressure [ dbar ]'); ax2.set_ylabel('Probability Density
ax2.axvline(x=meanp10,color='cyan',linestyle='-.',label='2010 Mean')
ax2.axvline(x=meanp20,color='orange',linestyle='-.',label='2020 Mean')
ax2.legend(loc='best',fontsize=14)

plt.show()
```







All there!

Q6: Variance reduction by averaging

As a bonus, can you demonstrate that the sample variance of N -sample averages decreases as N as we learned in class?

In [ ]:

```
# not enough time! :(
```

In [ ]:

```
# was messing around with trying to download multiple variables iterativ

def pier_data_download(years,vars_of_interest):
    # years should be an array or list of (number) years of interest
    # vars_of_interest should be a string list of variables to download

    year_string = [str(yy) for yy in years]
    num_vars = len(vars_of_interest)

    url_base = 'http://thredds.sccoos.org/thredds/dodsC/autoss/scripps_p
    urls = [url_base+yy+'.nc' for yy in year_string]

    variables = xr.DataArray([])
    df = np.array((1,num_vars)) # this will be the # of columns of the n
    for nn,vv enumerate(vars_of_interst):



years = np.arange(2005,2022) # going from 2005 to 2021
year_string = [str(yy) for yy in years]

url_base = 'http://thredds.sccoos.org/thredds/dodsC/autoss/scripps_pier-
urls = [url_base+yy+'.nc' for yy in year_string]
time = np.array([])
temp = np.array([])
sal = np.array([])
p = np.array([])
chl = np.array([])
station = np.array([])
lon = np.array([]); lat = np.array([])
zeta = np.array([])

for n,fn in enumerate(urls):
    # read current file:
    nc = netCDF4.Dataset(fn)
    t_now = nc['time'][:]
    t_here = nc['temperature'][:]
    s_here = nc['salinity'][:]
    p_here = nc['pressure'][:]
    c_here = nc['chlorophyll'][:]
    s_here = nc['station'][:]
    lon_here = nc['lon'][:]; lat_here = nc['lat'][:]
    z_here = nc['depth'][:]

    # append to extended record
    time = np.append(time,t_now)
    temp = np.append(temp,t_here)
    sal = np.append(sal,s_here)
    p = np.append(p,p_here)
    chl = np.append(chl,c_here)
    station = np.append(station,s_here)
    lon = np.append(lon,lon_here); lat = np.append(lat,lat_here)
    zeta = np.append(zeta,z_here)

s0 = dt.datetime(1970,1,1)
dates = [s0+dt.timedelta(seconds=float(tt)) for tt in time.data]

# trying to make a dataframe with all data, but noticed that matplotlib 
SP = pd.DataFrame({'dates':np.array(dates),'temp':temp,'p':p,'chl':chl})
# information I couldn't add because size requirements: 'sal':sal, 'stat
```