

C3S Data Rescue Service - Unit Conversion Tools

This is a library containing collection of routines that have been developed to assist climate data rescuers in historical units conversions.

Getting started

Python is required. The original fortran format for these routines can be found here. And a perl implementation can be found here.

Install the package

git clone git://github.com/C3S-Data-Rescue-Lot1-WP3/CS3-DR-Unit_Conversions

In this wiki you will find one page for each function (see panel on the right).

Typical workflow

Climate data are often typed into a spreadsheet that closely resembles the structure of the source (e.g., a weather log-book). Since the data structure is different for each source, it is usually not possible to work with large amounts of data until they are converted into a common machine-readable format (such as the SEF or IMMA).

As an example we have used the Jamestown 1886 from oldweather3 the logbook data was stored in two separate files one for ship positions at midnight/noon and the other for hourly observations:

Transcribed Data (that has been quality controlled for date/location and Observation accuracy) * positions.qc.out.revised - ascii fixed field format of ship locations at noon. * obs.qc.out.revised - ascii fixed file field format of hourly ship observations (temp,pressure)

Conversion Script (to_imma.py) * apply conversions if necessary using lmrllib.py (time/barometric/temperature/wind) * add position only obs (filling in gaps) * convert dates to UTC (extra caution at dateline boundary oldweather3 data assumes that the ships logs reflect correct hemisphere)

Converted Observations * imma.jamestown

The **bar_Eng_in2mb** function performs the barometric conversion for a single variable from units of inches (English) Hg (English units of inches of Mercury) to units of millibars (mb) for barometric pressure.

- **input(barometric pressure in English inches of Mercury)**
 - **output(barometric pressure in millibars)**
-

The **bar_Fr_in2mb** function performs the barometric conversion for a single variable from French units of inches of Mercury to units of millibars (mb) for barometric pressure.

- **input(barometric pressure in French inches of Mercury)**
- **output(barometric pressure in millibars)**

The **baro_G_correction** function calculates the barometric correction for gravity for a single variable. It is important to note that the correction value is dependent upon the latitude of the observation, and there is no conversion of units involved in the specific task that this routine performs.

adopt gmode=1 for 1955-forward, and gmode=2 for data prior to 1955):

	g1 (equation 1)	g2 (equation 2)	Comment
1 =	g45	g0	yields List (1966), Table 47B
2 =	g0	g0	follows GRAVCOR (pre-1955)
3 =	g45	g45	(of unknown utility)

Conventions: g45 is a "best" estimate of acceleration of gravity at 45 deg latitude and sea level, and g0 is the value of standard (normal) gravity "to which reported barometric data in mm or inches of mercury shall refer, but it does not represent the value of gravity at latitude 45 deg, at sea level"

- **input(barometric pressure,latitude,gmode)**
- **output(Gravity Correction)**

The **baro_mb2Eng_in** performs the barometric conversion for a single variable from units of millibars (mb) to English units of inches of Mercury for barometric pressure.

- **input(barometric pressure in millibars)**
- **output(barometric pressure in English inches of Mercury)**

The **baro_mb2Fr_in** function performs the barometric conversion for a single variable from units of millibars (mb) to French units of inches of Mercury for barometric pressure.

- **input(barometric pressure in millibars)**
- **output(barometric pressure in French inches of Mercury)**

The **baro_mb2mm** function performs the barometric conversion from units of millibars (mb) to units of millimeters of Mercury (mm Hg) for barometric pressure.

- **input(barometric pressure in millibars)**
 - **output(barometric pressure in millimeters of Mercury)**
-

The **baro_mm2mb** function performs the barometric conversion for a single variable from units of millimeters of Mercury (millimeters Hg) to units of millibars (mb) for barometric pressure.

- **input(barometric pressure in millimeters of Mercury)**
 - **output(barometric pressure in millibars)**
-

The **baro_tempC_correction** function calculates the barometer temperature correction given a temperature value in degrees Celsius and barometric pressure in mm or mb.

- **input(Barometric Pressure in mm or mb, Temperature in degrees Celsius)**
 - **output(Correction value of Barometric Pressure)**
-

The **baro_tempF_correction** function calculates the barometric temperature correction given a temperature in degrees Fahrenheit and barometric pressure given in English inches of Mercury.

- **input(Barometric Pressure in English inches of Mercury, Temperature in degrees Fahrenheit)**
 - **output(Correction value of barometric pressure)**
-

The **baro_temp_correction** performs the temperature correction for a generalized temperature value, regardless of whether the units associated with the temperature value are given in degrees Fahrenheit or degrees Celsius. It is important to note that there is no conversion of units involved in the specific task that this routine performs.

u	bp	t	of scale (ts)	of mercury (th)
-	-----	-----	-----	-----
0	mm or mb	Celsius	0C	0C
1	Eng. in.	Fahrenheit	62F (16.667C)	32F (0C) (pre-1955)
2	Eng. in.	Fahrenheit	32F (0C)	32F (0C) (1955-)
3	French in.	Reaumur	13R (16.25C)	0R (0C)

- **input(Barometric Pressure, Temperature, u)**
 - **output(Correction value of Barometric Pressure)**
-

The **cloud_tenths_clear2oktas** function performs the cloud conversion for a single variable from tenths (of sky clear) to oktas (the cloud cover present in the sky in units of eighths).

- **input(tenths of clear sky)**
 - **output(eighths of cloud coverage)**
-

The **cloud_tenths_covered2oktas** function performs the cloud conversion for a single variable from tenths (of sky covered) to oktas (eighths of sky covered).

- **input(tenths of sky covered)**
 - **output(eighths of sky covered)**
-

The **temp_c2** function performs the temperature conversion for a single variable from units of degrees Celsius to units of degrees Fahrenheit for temperature.

- **input(Temperature in Celsius)**
 - **output(Temperature in Fahrenheit)**
-

The **temp_c2k** function performs the temperature conversion for a single variable from units of degrees Celsius to units of degrees Kelvins for temperature.

- **input(Temperature in Celsius)**
 - **output(Temperature in Kelvin)**
-

The **temp_c2r** function performs the temperature conversion for a single variable from units of degrees Celsius to units of degrees Reaumur for temperature.

- **input(Temperature in Celsius)**
 - **output(Temperature in Reaumur)**
-

The **temp_f2c** function performs the temperature conversion for a single variable from units of degrees Fahrenheit to units of degrees Celsius for temperature.

- **input(Temperature in Fahrenheit)**
 - **output(Temperature in Celsius)**
-

The **temp_k2c** function performs the temperature conversion for a single variable from units of degrees Kelvins to units of degrees Celsius for temperature.

- **input(Temperature in Kelvin)**
 - **output(Temperature in Celsius)**
-

The **temp_r2c** function performs the temperature conversion for a single variable from units of degrees Reaumur to units of degrees Celsius for temperature.

- **input(Temperature in Reaumur)**
 - **output(Temperature in Celsius)**
-

The **time_date2julianday** function performs the time conversion from the date specified in the format of (iday,imonth,iyear) to the number of days since the date of 1 Jan 1770 (which is known as the Julian Day)

- **input(day, month, year)**
 - **output(number of days since 1 January 1770)**
-

The **time_julianday2date** function performs the time conversion from the number of days since the date of 1 Jan 1770 (which is known as the “Julian Day”), to the calendar date in the format of (iday,imonth,iyear)

- **input(universal time hour, universal time day)**
 - **output(iday,imonth,iyear)**
-

The **time_local_hour_julianday2UTC** function performs the time conversion from local standard hour and “Julian Day” to the time unit of UTC (universal coordinated time) for a given East longitude.

- **input(local standard hour in hundredths, Julian Day, east longitude in hundredths of degrees)**
 - **output(universal time hour, universal time day)**
-

The **wind_4chardir2deg** function performs the wind direction conversion for a single variable from the 4-character 32-point wind direction abbreviation c32 into degrees from 0 North. The 4-character 32-point wind direction abbreviation, is converted into both numeric code (from a range of 1-32, with 1 being the minimum value and 32 being the maximum value for the code) and units of degrees for wind direction. Where imiss is the value to return if no match is found.

- **input(Wind Direction given in C32 abbreviation (c32,dc,imiss))**
 - **output(Wind Direction in degrees clockwise 0 North or imiss)**
-

The **wind_Beaufort2kts** function performs the wind velocity conversion for a single variable from units of the Beaufort force, where the Beaufort force is represented by an integer from a range of 0-12 (with 0 being the minimum value for the force and 12 being the maximum value for the force), to units of knots

(kts), and more specifically, to the “old” (WMO code 1100) midpoint in units of knots (kts).

- **input(Beaufort force)**
 - **output(Wind velocity in knots)**
-

The **wind_Beaufort2mps** function performs the wind velocity conversion for a single variable from units of the Beaufort force, where the Beaufort force is represented by an integer from a range of 0-12 (with 0 being the minimum value for the force and 12 being the maximum value for the force), to units of meters per second (m/s), and more specifically, to the “old” (WMO code 1100) midpoint in units of meters per second (m/s).

- **input(Beaufort force)**
 - **output(Wind velocity in meters per second)**
-

The **wind_a_kts2mps** function performs the wind velocity conversion for a single variable from Admiralty nautical mile knots to units of meters per second (m/s).

- **input(Wind velocity in admiralty nautical mile knots)**
 - **output(Wind velocity in meters per second)**
-

The **wind_dircode2deg** function performs the wind direction conversion for a single variable from the 32-point wind direction code into degrees for the variable of wind direction. Where imiss is the value to return if no match is found.

- **input(Wind direction code (0-32), imiss)**
 - **output(Wind direction degrees from 0 North)**
-

The **wind_kts2mps** function performs the wind velocity conversion for a single variable from units of knots to units of meters per second (m/s) (in reference to the international nautical mile for the units of knots).

- **input(wind velocity in knots)**
 - **output(wind velocity in meters per second)**
-

The **wind_mps2a_kts** function performs the wind velocity conversion for a single variable from units of meters per second (m/s) to units of knots (in reference to the Admiralty nautical mile for the units of knots).

- **input(Wind velocity in meters per second)**
- **output(Wind velocity in Admiralty nautical mile knots)**

The **wind_mps2kts** function performs the wind velocity conversion for a single variable from units of meters per second (m/s) to units of knots (in reference to the international nautical mile for the units of knots).

- **input(wind velocity in meters per second)**
- **output(wind velocity in knots)**

The **wind_mps2us_kts** function performs the wind velocity conversion for a single variable from units of meters per second (m/s) to units of knots (in reference to the United States nautical mile for the units of knots).

- **input(Wind Velocity in meters per second)**
- **output(Wind Velocity in United States nautical mile knots)**

This function performs the wind velocity conversion for a single variable from units of knots to units of meters per second (m/s) (in reference to the United States nautical mile for the units of knots).

- **input(Wind Velocity in units of United States nautical mile knots)**
- **output(Wind Velocity in meters per second)**

The **wind_uv2dir** function performs the wind conversion for a single variable from the vector components (that is, from the u component and the v component for wind to wind direction, where direction is expressed in units of degrees clockwise from 0 degrees North.

- **input(Zonal wind, Meridional wind)**
- **output(Wind Direction in degrees clockwise from North)**

The **wind_uv2vel** function performs a wind conversion for a single variable from the vector components (that is, from the u component and the v component of the wind to wind velocity.

- **input(Zonal Wind, Meridional Wind)**
 - **output(Wind Velocity)**
-