```
https://medium.com/the-investors-handbook/coding-understanding-the-aroon-oscillator
-in-python-c3accb782d69
def aroon(Data, period, close, where):
    # Adding Columns
    Data = adder(Data, 10)
    # Max Highs
    for i in range(len(Data)):
        try:
            Data[i, where] = max(Data[i - period + 1:i + 1, 1])
        except ValueError:
            pass
    # Max Lows
    for i in range(len(Data)):
        try:
            Data[i, where + 1] = min(Data[i - period + 1:i + 1, 2])
        except ValueError:
            pass
    # Where the High Equals the Highest High in the period
    for i in range(len(Data)):
        if Data[i, 1] == Data[i, where]:
            Data[i, where + 2] = 1
    # Where the Low Equals the Lowest Low in the period
    for i in range(len(Data)):
        if Data[i, 2] == Data[i, where + 1]:
            Data[i, where + 3] = 1
    # Jumping Rows
    Data = jump(Data, period)
    # Calculating Aroon Up
    for i in range(len(Data)):
        try:
            try:
```

```
x = max(Data[i - period:i, 1])
                y = np.where(Data[i - period:i, 1] == x)
                y = np.array(y)
                distance = period - y
                Data[i - 1, where + 4] = 100 *((period - distance) / period)
except ValueError:
                pass
        except IndexError:
            pass
    # Calculating Aroon Down
    for i in range(len(Data)):
        try:
            try:
                x = min(Data[i - period:i, 2])
                y = np.where(Data[i - period:i, 2] == x)
                y = np.array(y)
                distance = period - y
                Data[i - 1, where + 5] = 100 *((period - distance) / period)
except ValueError:
                pass
        except IndexError:
            pass
    # Cleaning
    Data = deleter(Data, 5, 4)
    return Data
my_data[:, where_aroon_osc] = my_data[:, where_aroon_up] -
                              my_data[:, where_aroon_down]
```