Homework 10

- Scientific computing with Numpy and Matplotlib, unittests and interview exercises

Given April 16th – Due May 5th

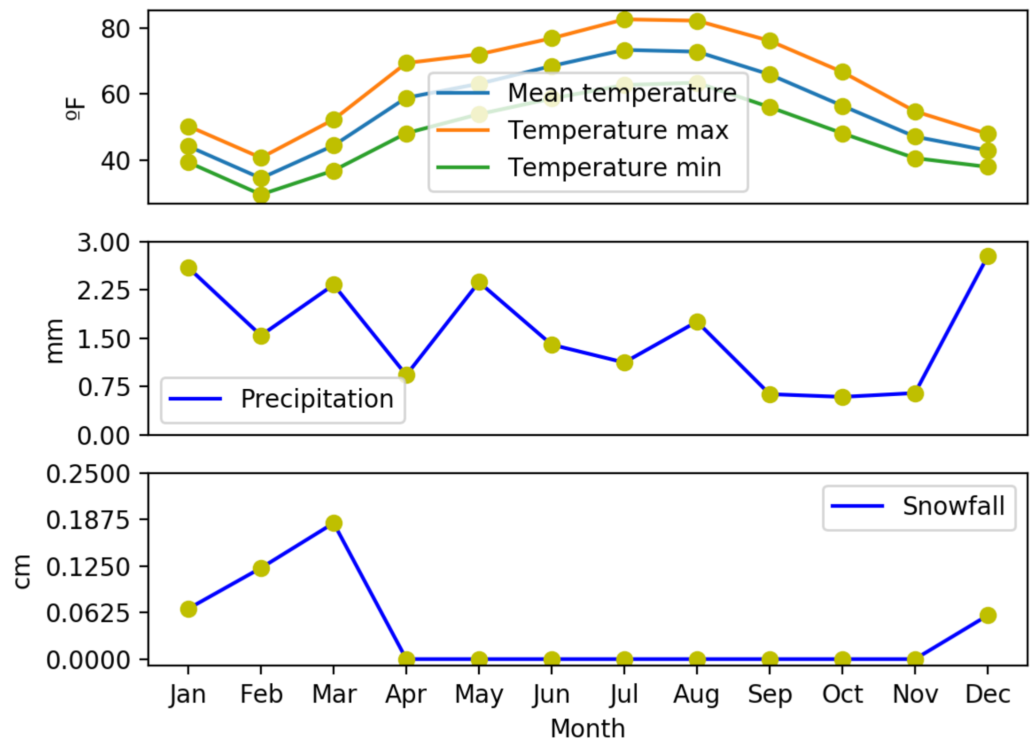
Exercise 1. Plot information about last year

In the *history_report_colorado_springs_2018.csv* file you have the following information for every day of that year:

- Mean temperature
- Total precipitation
- Snowfall amount
- Temperature daily max
- Temperature daily min

where temperatures are in ºF, precipitation in mm and snowfall in cm.

For this exercise you have to read the csv file and plot the data to visualize it. You may plot it as the following figure:



Tips:

- You can create a three-dimensional array using numpy:

  ```
  data = np.empty((12, 5, 31))
  ```

  Where 12 are for the months, 5 are for the number of parameters and 31 are for the number of days for each month.
- Then set all the numbers to NaN.

  ```
  data[:] = np.nan
  ```

- Then access to the CSV file and assign the corresponding values to the data array.
- Finally, you can compute the mean of a month as (this case is for the precipitation parameter):

  ```
  [np.nanmean(data[i, :, 1]) for i in range(12)]
  ```

  np.nanmean() method will compute the arithmetic mean along the specified axis, ignoring NaNs.

Exercise 2. Create a complete interview exercise

For this exercise you are going to create a complete coding interview exercise. A similar thing that the interviewers of large companies use to examine their candidates. You have to create three files:

- **run_empty.py** → This is the file you will give to the interviewed person. You should have a multiline comment explain the problem and some input/output examples to the function. Also, you should have the function the interviewed person has to code as empty. For example:

```
"""
Problem: Write a Python function that checks whether a passed
string is palindrome or not.
Note: A palindrome is a word, phrase, or sequence that reads the
same backward as forward.

Example 1   Input: “madam”
            Output: True

Example 2   Input: “Hello world!”
            Output: False

Example 3   Input: “Eva, Can I Stab Bats In A Cave?”
            Output: True
"""
```

```python
def is_palindrome(var):
    # Your code here
    pass
```

- **test.py** → This is the test file that will call the interviewed code to test the method performance. For example:

```python
from string_problems.palindrome.run import is_palindrome
import unittest

class Test(unittest.TestCase):

    def test_1(self):
        self.assertEqual(is_palindrome('madam'), True, "Should be 
True")

    def test_2(self):
        self.assertEqual(is_palindrome('redder'), True, "Should be 
True")

    def test_3(self):
        self.assertEqual(is_palindrome('race car'), True, "Should 
be True")

    def test_4(self):
        self.assertEqual(is_palindrome('Eva, Can I Stab Bats In A 
Cave?'), True, "Should be True")

# ...

    def test_9(self):
        self.assertEqual(is_palindrome('Bob'), True, "Should be 
True")

    def test_10(self):
        with self.assertRaises(AssertionError):
            is_palindrome(9)


if __name__ == '__main__':
    unittest.main()
```

- **run.py** → Like *run_empty.py* but with your own implementation of the function. All your tests should pass with the implementation in this file. This should be the most optimal way to solve it, a reference for the interviewer.

```python
import re

def is_palindrome(var):
    assert isinstance(var, str), 'The input should be a string'
    namesRegex = re.compile(r'''[^a-zA-Z]''')
    var = namesRegex.sub('', var)
    var = var.lower()
    return var == var[::-1]
```

NOTE: Of course, you canNOT do the palindrome problem for this exercise. Think or search for a problem and implement it the way is asked in this exercise.

If the exercise does not say which name to save the file. Submit your code files as *hm10_name_surname_ex_num.py*, where *num* is the exercise number 1, 2, etc. Comment everything so we know you wrote the code! On top of your files write this multiline comment with your information:

"""

Homework 1, Exercise 1 (or 2…)

Name

Date

Description of your program.

"""