

# Python model

## Steps in running the model in python (started 200526)

- `model.run_model(state, region)`
- `daily_weather = get_daily_weather(state, region)`
  - returns 1981-2020/week22
- `weekly_weather = calculate_weekly_weather(daily_weather)`
  - returns 1981-2020/week21
- `seasonal_weather = calculate_seasonal_weather(weekly_weather)`
  - returns 1981-2021

```
• 2020: {'fall': {'degree_week_season': 0.0,
                 'precip_season': 0.0,
                 'temp_season': 0.0},
        'spring': {'degree_week_season': 0.0,
                  'precip_season': 2.5718597719873264,
                  'temp_season': 10.7141171619569},
        'summer': {'degree_week_season': 0.0,
                  'precip_season': 0.0,
                  'temp_season': 0.0},
        'winter': {'degree_week_season': 25.020196358071754,
                  'precip_season': 2.4027636876205745,
                  'temp_season': 3.1947348318693845}}
```

- `weekly_normals = calculate_weekly_normals(weekly_weather)`
  - return reasonable
- `seasonal_normals = calculate_seasonal_normals(weekly_normals)`

```
• {'fall': {'degree_week_season_norm': 32.28328949055344,
           'precip_season_norm': 2.036040032318641,
           'temp_season_norm': 14.050616168374207},
   'spring': {'degree_week_season_norm': 0.0,
             'precip_season_norm': 2.5237899283778593,
             'temp_season_norm': 13.218438315013097},
   'summer': {'degree_week_season_norm': 15.339247600693605,
             'precip_season_norm': 1.8008227971626916,
             'temp_season_norm': 24.383073424050902},
   'winter': {'degree_week_season_norm': 9.933319843247213,
             'precip_season_norm': 1.9007136782741765,
             'temp_season_norm': 1.0790843877313672}}
```

- not good?
- `weekly_diffs = calculate_weekly_diffs(weekly_weather, weekly_normals)`
  - returns 2005 - 2020/week21

- 2020: {1: {'degree\_week\_diff': 0, 'precip\_week\_diff': 2.308374367823269, 'temp\_week\_diff': 5.682186852568982}, 2: {'degree\_week\_diff': 0, 'precip\_week\_diff': 6.989623108931939, 'temp\_week\_diff': 5.84392805275305}, 3: {'degree\_week\_diff': 0, 'precip\_week\_diff': 1.840563673191461, 'temp\_week\_diff': 4.565747032636812}, 4: {'degree\_week\_diff': 0, 'precip\_week\_diff': -0.21884846160374805, 'temp\_week\_diff': -0.893717392535004}, 5: {'degree\_week\_diff': 0, 'precip\_week\_diff': -1.6230414652298615, 'temp\_week\_diff': 0.9424525952094789}, 6: {'degree\_week\_diff': 0, 'precip\_week\_diff': 1.4695358466737318, 'temp\_week\_diff': 5.370844795472097}, 7: {'degree\_week\_diff': 0, 'precip\_week\_diff': 1.880304966371984, 'temp\_week\_diff': -1.0578861978476652}, 8: {'degree\_week\_diff': 0, 'precip\_week\_diff': -1.1605559161663699, 'temp\_week\_diff': -1.5443196284077159}, 9: {'degree\_week\_diff': 0, 'precip\_week\_diff': -0.09366844294108256, 'temp\_week\_diff': -0.5794312337403582}, 10: {'degree\_week\_diff': 0, 'precip\_week\_diff': -0.003445720652338169, 'temp\_week\_diff': 1.7032236296033094}, 11: {'degree\_week\_diff': 0, 'precip\_week\_diff': 1.188863109072992, 'temp\_week\_diff': 2.4080467602563305}, 12: {'degree\_week\_diff': 0, 'precip\_week\_diff': 4.637077095073915, 'temp\_week\_diff': 1.060789083046882}, 13: {'degree\_week\_diff': 0, 'precip\_week\_diff': -1.3215119621992675, 'temp\_week\_diff': -0.2656845970759285}, 14: {'degree\_week\_diff': 0, 'precip\_week\_diff': -0.7376526533723604, 'temp\_week\_diff': 1.3677699099142089}, 15: {'degree\_week\_diff': 0, 'precip\_week\_diff': -1.4553916893702714, 'temp\_week\_diff': 1.191005520088968}, 16: {'degree\_week\_diff': 0, 'precip\_week\_diff': -1.6130754073986175, 'temp\_week\_diff': -6.403501944567171}, 17: {'degree\_week\_diff': 0, 'precip\_week\_diff': 0.16515468455716276, 'temp\_week\_diff': -2.8888956428204846}, 18: {'degree\_week\_diff': 0, 'precip\_week\_diff': 1.1540223358628956, 'temp\_week\_diff': -1.8941569807668248}, 19: {'degree\_week\_diff': 0, 'precip\_week\_diff': 0.6052393742837743, 'temp\_week\_diff': -3.083952187230345}, 20: {'degree\_week\_diff': 0, 'precip\_week\_diff': -1.3968075225831864, 'temp\_week\_diff': -4.50203879926571}, 21: {'degree\_week\_diff': 0, 'precip\_week\_diff': 2.0896532708667452, 'temp\_week\_diff': -0.7827753087200158}}}

***Follows fortran program***

1. load epi week finder
  - a. Used to convert YYYY-MM-DD to EPI Year, Week, DOW
2. Load weather data from 110x
3. Convert to dictionary
  - a. convert dates to epi week
  - b. average temp max and temp min
  - c. convert fahrenheit to celsiu
  - d. convert inches to cm
  - e. Dictionary structure
    - i. [year][week]['daily']
      1. contains daily precip and temp\_average
4. Calculate weekly values of temp and precip
  - a. precip - add up total for week
  - b. degree week
    - i.  $((\text{Sum daily averages for week})/7) - 22.0 = dw$ 
      1. if  $dw < 0$ ; degree week = 0
      2. else; degree week = dw
  - c. Dictionary structure
    - i. [year][week]['weekly']
5. Calculate seasonal averages **wonder if this should be the last week and not the average?**
  - a. winter = weeks [1,13], spring [14, 26], etc.
    - i.  $(\text{sum weekly degree week})/13$
6. Calculate seasonal 30 year normals
  - a.  $(\text{sum 30 year season values})/30$
7. Calculate weekly 30 year normals
  - a.  $(\text{sum 30 year week values})/30$
8. Calculate Abnormal values for year of model