Data Analytics

January 26, 2021

```
In [2]: import numpy as np

        ary1 = np.array([1,2,3])

        print(ary1)
```

```
[1 2 3]
```

```
In [3]: print(type(ary1))
```

```
<class 'numpy.ndarray'>
```

```
In [4]: print(ary1.shape)
```

```
(3,)
```

```
In [5]: ary1[2]
```

```
Out[5]: 3
```

```
In [7]: ary1[2]=5
        ary1
```

```
Out[7]: array([1, 2, 5])
```

```
In [9]: ary2 = np.array([[1,2,3],[3,4,5]])
        ary2
```

```
Out[9]: array([[1, 2, 3],
               [3, 4, 5]])
```

```
In [10]: ary2[1][2]
```

```
Out[10]: 5
```

```
In [11]: print(ary2.shape)
```

```
(2, 3)
```

```
In [12]: ary2[1][-1]
```

```
Out[12]: 5
```

```
In [13]: ary2[1][-3]
```

```
Out[13]: 3
```

```
In [14]: arys = np.array(['Chaina', 'India','Canada','USA'])
         arys
```

```
Out[14]: array(['Chaina', 'India', 'Canada', 'USA'], dtype='<U6')
```

```
In [15]: aryR = np.arange(0,20,2)
```

```
In [87]: aryR = np.arange(0,30,3)
         aryR
```

```
Out[87]: array([ 0,  3,  6,  9, 12, 15, 18, 21, 24, 27])
```

```
In [18]: aryL = np.linspace(0,10,20)
         aryL
```

```
Out[18]: array([ 0.        ,  0.52631579,  1.05263158,  1.57894737,  2.10526316,
                 2.63157895,  3.15789474,  3.68421053,  4.21052632,  4.73684211,
                 5.26315789,  5.78947368,  6.31578947,  6.84210526,  7.36842105,
                 7.89473684,  8.42105263,  8.94736842,  9.47368421, 10.        ])
```

```
In [19]: ary = np.random.rand(10)
         ary
```

```
Out[19]: array([0.56750196, 0.60980437, 0.16004783, 0.69149488, 0.29569082,
                0.62240006, 0.21497158, 0.12443889, 0.22541522, 0.47230696])
```

```
In [20]: ary = np.random.rand(3,4)
         ary
```

```
Out[20]: array([[0.58403665, 0.9600567 , 0.97892197, 0.54657973],
                [0.3440654 , 0.18368897, 0.3425167 , 0.39620669],
                [0.44122453, 0.54291041, 0.69028765, 0.06868475]])
```

```
In [21]: print(np.zeros(10))
         print('/n')
         print(np.zeros((2,3)))
```

```
[0. 0. 0. 0. 0. 0. 0. 0. 0. 0.]
/n
[[0. 0. 0.]
 [0. 0. 0.]]
```

```
In [22]: print(np.ones((10,10,3)))
```

```
[[[1. 1. 1.]
  [1. 1. 1.]
  [1. 1. 1.]
  [1. 1. 1.]
  [1. 1. 1.]
  [1. 1. 1.]
  [1. 1. 1.]
  [1. 1. 1.]
  [1. 1. 1.]
  [1. 1. 1.]]

 [[1. 1. 1.]
  [1. 1. 1.]
  [1. 1. 1.]
  [1. 1. 1.]
  [1. 1. 1.]
  [1. 1. 1.]
  [1. 1. 1.]
  [1. 1. 1.]
  [1. 1. 1.]
  [1. 1. 1.]]

 [[1. 1. 1.]
  [1. 1. 1.]
  [1. 1. 1.]
  [1. 1. 1.]
  [1. 1. 1.]
  [1. 1. 1.]
  [1. 1. 1.]
  [1. 1. 1.]
  [1. 1. 1.]
  [1. 1. 1.]]

 [[1. 1. 1.]
  [1. 1. 1.]
  [1. 1. 1.]
  [1. 1. 1.]
  [1. 1. 1.]
  [1. 1. 1.]
  [1. 1. 1.]
  [1. 1. 1.]
  [1. 1. 1.]
  [1. 1. 1.]]

 [[1. 1. 1.]
  [1. 1. 1.]
```

```
 [1. 1. 1.]
 [1. 1. 1.]
 [1. 1. 1.]
 [1. 1. 1.]
 [1. 1. 1.]
 [1. 1. 1.]
 [1. 1. 1.]
 [1. 1. 1.]]

[[1. 1. 1.]
 [1. 1. 1.]
 [1. 1. 1.]
 [1. 1. 1.]
 [1. 1. 1.]
 [1. 1. 1.]
 [1. 1. 1.]
 [1. 1. 1.]
 [1. 1. 1.]
 [1. 1. 1.]]

[[1. 1. 1.]
 [1. 1. 1.]
 [1. 1. 1.]
 [1. 1. 1.]
 [1. 1. 1.]
 [1. 1. 1.]
 [1. 1. 1.]
 [1. 1. 1.]
 [1. 1. 1.]
 [1. 1. 1.]]

[[1. 1. 1.]
 [1. 1. 1.]
 [1. 1. 1.]
 [1. 1. 1.]
 [1. 1. 1.]
 [1. 1. 1.]
 [1. 1. 1.]
 [1. 1. 1.]
 [1. 1. 1.]
 [1. 1. 1.]]

[[1. 1. 1.]
 [1. 1. 1.]
 [1. 1. 1.]
 [1. 1. 1.]
 [1. 1. 1.]
 [1. 1. 1.]
```

```
  [1. 1. 1.]
  [1. 1. 1.]
  [1. 1. 1.]
  [1. 1. 1.]]

 [[1. 1. 1.]
  [1. 1. 1.]
  [1. 1. 1.]
  [1. 1. 1.]
  [1. 1. 1.]
  [1. 1. 1.]
  [1. 1. 1.]
  [1. 1. 1.]
  [1. 1. 1.]
  [1. 1. 1.]]]
```

```
In [23]: ary = [0,1,2]
         print(np.repeat(ary, 5))
```

```
[0 0 0 0 0 1 1 1 1 1 2 2 2 2 2]
```

```
In [24]: print(np.tile(ary,5))
```

```
[0 1 2 0 1 2 0 1 2 0 1 2 0 1 2]
```

```
In [25]: identity_matrix = np.eye(3)
         identity_matrix
```

```
Out[25]: array([[1., 0., 0.],
                [0., 1., 0.],
                [0., 0., 1.]])
```

```
In [26]: np.diag([1,2,3,4,5])
```

```
Out[26]: array([[1, 0, 0, 0, 0],
                [0, 2, 0, 0, 0],
                [0, 0, 3, 0, 0],
                [0, 0, 0, 4, 0],
                [0, 0, 0, 0, 5]])
```

```
In [27]: ary = np.random.rand(5,5)
         ary
```

```
Out[27]: array([[0.25226077, 0.2579296 , 0.46973991, 0.87427238, 0.44335414],
                [0.18210193, 0.41776999, 0.87948047, 0.36747753, 0.07804553],
                [0.39424446, 0.03643247, 0.83427572, 0.39335018, 0.60132669],
                [0.59417161, 0.0712503 , 0.98424964, 0.20258022, 0.33971214],
                [0.6561921 , 0.87046826, 0.86435258, 0.74885978, 0.60192395]])
```

```
In [28]: np.diag(ary)

Out[28]: array([0.25226077, 0.41776999, 0.83427572, 0.20258022, 0.60192395])

In [29]: ary.ndim

Out[29]: 2

In [30]: ary.size

Out[30]: 25

In [31]: ary.shape

Out[31]: (5, 5)

In [32]: np.random.randint(-10,10,5)

Out[32]: array([-1,  1, -5, -2,  6])

In [34]: ary/4

Out[34]: array([[0.06306519, 0.0644824 , 0.11743498, 0.2185681 , 0.11083854],
                [0.04552548, 0.1044425 , 0.21987012, 0.09186938, 0.01951138],
                [0.09856111, 0.00910812, 0.20856893, 0.09833755, 0.15033167],
                [0.1485429 , 0.01781257, 0.24606241, 0.05064505, 0.08492804],
                [0.16404802, 0.21761706, 0.21608814, 0.18721495, 0.15048099]])

In [35]: np.exp(ary)

Out[35]: array([[1.28693158, 1.2942477 , 1.59957811, 2.39713046, 1.55792396],
                [1.19973648, 1.51857135, 2.40964749, 1.44408735, 1.08117189],
                [1.4832631 , 1.03710426, 2.30314532, 1.48193725, 1.82453779],
                [1.81152968, 1.07384997, 2.67580332, 1.22455831, 1.40454323],
                [1.92743884, 2.3880288 , 2.37346895, 2.11458755, 1.82562784]])

In [36]: ary

Out[36]: array([[0.25226077, 0.2579296 , 0.46973991, 0.87427238, 0.44335414],
                [0.18210193, 0.41776999, 0.87948047, 0.36747753, 0.07804553],
                [0.39424446, 0.03643247, 0.83427572, 0.39335018, 0.60132669],
                [0.59417161, 0.0712503 , 0.98424964, 0.20258022, 0.33971214],
                [0.6561921 , 0.87046826, 0.86435258, 0.74885978, 0.60192395]])

In [38]: np.log2(ary)

Out[38]: array([[-1.98701225, -1.95495074, -1.09006592, -0.19384527, -1.17346854],
                [-2.45718189, -1.25921923, -0.18527656, -1.44427205, -3.67954015],
                [-1.34283762, -4.77863152, -0.26140383, -1.34611384, -0.7337791 ],
                [-0.75104841, -3.8109602 , -0.02290381, -2.30343481, -1.5576153 ],
                [-0.60780988, -0.20013641, -0.21030818, -0.41723248, -0.73234687]])
```

```
In [39]: np.sin(ary)

Out[39]: array([[0.24959381, 0.25507919, 0.45265438, 0.7670769 , 0.42897173],
                [0.18109715, 0.40572324, 0.77040776, 0.35926252, 0.07796633],
                [0.38411072, 0.03642441, 0.7408102 , 0.38328489, 0.56573694],
                [0.55982254, 0.07119003, 0.83285701, 0.20119745, 0.3332157 ],
                [0.6101042 , 0.7646308 , 0.76067516, 0.68080403, 0.56622933]])

In [40]: np.sum(ary)

Out[40]: 12.415822346118937

In [43]: np.sum(ary, axis=0)

Out[43]: array([2.07897086, 1.65385061, 4.03209832, 2.5865401 , 2.06436246])

In [44]: np.max(ary)

Out[44]: 0.9842496432443214

In [45]: np.mean(ary)

Out[45]: 0.49663289384475745

In [46]: np.std(ary)

Out[46]: 0.28134296733775815

In [47]: np.var(ary)

Out[47]: 0.07915386527041485

In [49]: ary[1:,2:4]

Out[49]: array([[0.87948047, 0.36747753],
                [0.83427572, 0.39335018],
                [0.98424964, 0.20258022],
                [0.86435258, 0.74885978]])

In [51]: np.sort(ary)

Out[51]: array([[0.25226077, 0.2579296 , 0.44335414, 0.46973991, 0.87427238],
                [0.07804553, 0.18210193, 0.36747753, 0.41776999, 0.87948047],
                [0.03643247, 0.39335018, 0.39424446, 0.60132669, 0.83427572],
                [0.0712503 , 0.20258022, 0.33971214, 0.59417161, 0.98424964],
                [0.60192395, 0.6561921 , 0.74885978, 0.86435258, 0.87046826]])

In [52]: ary
```

```
Out[52]: array([[0.25226077, 0.2579296 , 0.46973991, 0.87427238, 0.44335414],
                [0.18210193, 0.41776999, 0.87948047, 0.36747753, 0.07804553],
                [0.39424446, 0.03643247, 0.83427572, 0.39335018, 0.60132669],
                [0.59417161, 0.0712503 , 0.98424964, 0.20258022, 0.33971214],
                [0.6561921 , 0.87046826, 0.86435258, 0.74885978, 0.60192395]])

In [53]: ary.T

Out[53]: array([[0.25226077, 0.18210193, 0.39424446, 0.59417161, 0.6561921 ],
                [0.2579296 , 0.41776999, 0.03643247, 0.0712503 , 0.87046826],
                [0.46973991, 0.87948047, 0.83427572, 0.98424964, 0.86435258],
                [0.87427238, 0.36747753, 0.39335018, 0.20258022, 0.74885978],
                [0.44335414, 0.07804553, 0.60132669, 0.33971214, 0.60192395]])

In [54]: ary[:3,:].T

Out[54]: array([[0.25226077, 0.18210193, 0.39424446],
                [0.2579296 , 0.41776999, 0.03643247],
                [0.46973991, 0.87948047, 0.83427572],
                [0.87427238, 0.36747753, 0.39335018],
                [0.44335414, 0.07804553, 0.60132669]])

In [55]: ary[:3,:].flatten()

Out[55]: array([0.25226077, 0.2579296 , 0.46973991, 0.87427238, 0.44335414,
                0.18210193, 0.41776999, 0.87948047, 0.36747753, 0.07804553,
                0.39424446, 0.03643247, 0.83427572, 0.39335018, 0.60132669])

In [59]: ary = np.array([4,5,6,7,8])
         ary

Out[59]: array([4, 5, 6, 7, 8])

In [61]: ary1 = np.append(ary,5)
         ary

Out[61]: array([4, 5, 6, 7, 8])

In [62]: ary2 = np.insert(ary, 0, [1,2,3])
         ary2

Out[62]: array([1, 2, 3, 4, 5, 6, 7, 8])

In [65]: ary3 = np.delete(ary2, 0)
         ary3

Out[65]: array([2, 3, 4, 5, 6, 7, 8])

In [67]: aryC = ary3.copy()
         aryC
```

```
Out[67]: array([2, 3, 4, 5, 6, 7, 8])

In [76]: ary1 = np.array([[1,2,3],[4,5,6]])
         ary2 = np.array([[4,5,6],[7,8,9]])
         ary3 = np.array([[8,7,6],[1,5,9]])
         ary_cat = np.concatenate((ary1,ary2,ary3),axis=1)
         print(ary_cat)

[[1 2 3 4 5 6 8 7 6]
 [4 5 6 7 8 9 1 5 9]]


In [77]: catV = np.vstack((ary1, ary2,ary3))
         catV

Out[77]: array([[1, 2, 3],
                [4, 5, 6],
                [4, 5, 6],
                [7, 8, 9],
                [8, 7, 6],
                [1, 5, 9]])

In [78]: catH = np.hstack((ary1, ary2,ary3))
         catH

Out[78]: array([[1, 2, 3, 4, 5, 6, 8, 7, 6],
                [4, 5, 6, 7, 8, 9, 1, 5, 9]])

In [80]: ary = np.array([1,5,3,6,7,4,245,1,7,45,7,56,36])
         np.unique(ary)

Out[80]: array([  1,   3,   4,   5,   6,   7,  36,  45,  56, 245])

In [85]: #intersection, differntiation, neither
         ary1 = np.array([1,2,3,4])
         ary2 = np.array([2,3,6,4])
         print(np.intersect1d(ary1, ary2))
         print(np.union1d(ary1, ary2))
         print(np.setdiff1d(ary2, ary1))
         print(np.setxor1d(ary1, ary2))

[2 3 4]
[1 2 3 4 6]
[6]
[1 6]


In [4]: import pandas as pd

In [88]: #for pandas
```

```
In [5]: import pandas as pd
        import numpy as np
        import seaborn as sns
        import matplotlib.pyplot as plt
        %matplotlib inline
        sns.set(color_codes=True)

In [9]: data = pd.read_csv('/home/user/GODAVARI/coOrdinate1.csv')
        data.head()

Out[9]:    Unnamed: 0  Albedo_inst  AvgSurfT_inst  CanopInt_inst  ECanop_tavg  \
        0           0    11.920000     308.211884            0.0         0.00
        1           0    11.890000     307.378479            0.0         0.00
        2           0    11.849998     307.798584            0.0         0.00
        3           0    11.820000     308.330750            0.0         0.04
        4           0    11.790000     307.115631            0.0         0.00

           ESoil_tavg  Evap_tavg  LWdown_f_tavg  Lwnet_tavg  PotEvap_tavg  ...  \
        0   18.570000   0.000082     340.456329 -167.500580    741.664856  ...
        1   18.219999   0.000081     335.034790 -166.838684    769.109375  ...
        2   16.670000   0.000078     332.093597 -170.586090    739.671814  ...
        3   15.900000   0.000076     339.926392 -165.926682    739.387756  ...
        4   16.160000   0.000075     331.670410 -165.583313    794.559509  ...

           SoilMoi40_100cm_inst  SoilTMP0_10cm_inst  SoilTMP100_200cm_inst  \
        0            154.382996          297.980682             297.915070
        1            153.203995          297.227661             297.881104
        2            152.104996          296.832367             297.839264
        3            151.041000          297.271484             297.801300
        4            149.955002          296.841766             297.758514

           SoilTMP10_40cm_inst  SoilTMP40_100cm_inst  Swnet_tavg  Tair_f_inst  \
        0           295.425629            296.595581  657.469971   303.633453
        1           295.138550            296.530579  662.510010   302.845917
        2           294.825287            296.441772  665.369995   302.986023
        3           294.642761            296.326202  658.780029   303.447296
        4           294.584229            296.230713  667.059998   302.692627

            Tveg_tavg  Wind_f_inst                        geometry
        0  186.119995     1.908393  <map object at 0x7f1b31fc6a90>
        1  183.759995     2.500216  <map object at 0x7f1b32105780>
        2  177.369995     1.803373  <map object at 0x7f1b321055f8>
        3  172.919998     1.902477  <map object at 0x7f1b31ff14a8>
        4  172.009995     3.501946  <map object at 0x7f1b32105a58>

        [5 rows x 38 columns]

In [10]: data.dtypes
```

```
Out[10]: Unnamed: 0                int64
         Albedo_inst             float64
         AvgSurfT_inst           float64
         CanopInt_inst           float64
         ECanop_tavg             float64
         ESoil_tavg              float64
         Evap_tavg               float64
         LWdown_f_tavg           float64
         Lwnet_tavg              float64
         PotEvap_tavg            float64
         Psurf_f_inst            float64
         Qair_f_inst             float64
         Qg_tavg                 float64
         Qh_tavg                 float64
         Qle_tavg                float64
         Qs_acc                  float64
         Qsb_acc                 float64
         Qsm_acc                   int64
         Rainf_f_tavg            float64
         Rainf_tavg              float64
         RootMoist_inst          float64
         SWE_inst                  int64
         SWdown_f_tavg           float64
         SnowDepth_inst            int64
         Snowf_tavg                int64
         SoilMoi0_10cm_inst      float64
         SoilMoi100_200cm_inst   float64
         SoilMoi10_40cm_inst     float64
         SoilMoi40_100cm_inst    float64
         SoilTMP0_10cm_inst      float64
         SoilTMP100_200cm_inst   float64
         SoilTMP10_40cm_inst     float64
         SoilTMP40_100cm_inst    float64
         Swnet_tavg              float64
         Tair_f_inst             float64
         Tveg_tavg               float64
         Wind_f_inst             float64
         geometry                 object
         dtype: object

In [11]: data.columns

Out[11]: Index(['Unnamed: 0', 'Albedo_inst', 'AvgSurfT_inst', 'CanopInt_inst',
                'ECanop_tavg', 'ESoil_tavg', 'Evap_tavg', 'LWdown_f_tavg', 'Lwnet_tavg',
                'PotEvap_tavg', 'Psurf_f_inst', 'Qair_f_inst', 'Qg_tavg', 'Qh_tavg',
                'Qle_tavg', 'Qs_acc', 'Qsb_acc', 'Qsm_acc', 'Rainf_f_tavg',
                'Rainf_tavg', 'RootMoist_inst', 'SWE_inst', 'SWdown_f_tavg',
                'SnowDepth_inst', 'Snowf_tavg', 'SoilMoi0_10cm_inst',
```

```
       'SoilMoi100_200cm_inst', 'SoilMoi10_40cm_inst', 'SoilMoi40_100cm_inst',
       'SoilTMP0_10cm_inst', 'SoilTMP100_200cm_inst', 'SoilTMP10_40cm_inst',
       'SoilTMP40_100cm_inst', 'Swnet_tavg', 'Tair_f_inst', 'Tveg_tavg',
       'Wind_f_inst', 'geometry'],
      dtype='object')
```

In [13]:
```
data.describe(include='all')
```

Out[13]:

	Unnamed: 0	Albedo_inst	AvgSurfT_inst	CanopInt_inst	ECanop_tavg
count	5843.0	5843.000000	5843.000000	5843.000000	5843.000000
unique	NaN	NaN	NaN	NaN	NaN
top	NaN	NaN	NaN	NaN	NaN
freq	NaN	NaN	NaN	NaN	NaN
mean	0.0	13.528428	312.764254	0.038741	18.791595
std	0.0	1.597749	8.716488	0.116089	48.560459
min	0.0	11.500000	294.225433	0.000000	0.000000
25%	0.0	12.320000	306.353516	0.000000	0.000000
50%	0.0	13.240000	309.525757	0.000000	0.000000
75%	0.0	14.150000	319.607880	0.001000	6.510000
max	0.0	17.000000	334.416504	0.500000	409.750000

	ESoil_tavg	Evap_tavg	LWdown_f_tavg	Lwnet_tavg	PotEvap_tavg
count	5843.000000	5843.000000	5843.000000	5843.000000	5843.000000
unique	NaN	NaN	NaN	NaN	NaN
top	NaN	NaN	NaN	NaN	NaN
freq	NaN	NaN	NaN	NaN	NaN
mean	66.761790	0.000077	412.222789	-127.603282	698.790264
std	63.233389	0.000062	38.467980	62.357138	231.067941
min	0.700000	0.000001	308.770874	-297.765778	25.336960
25%	12.155000	0.000019	381.936813	-176.656120	572.310974
50%	43.570000	0.000065	417.293640	-131.319382	716.754517
75%	111.680000	0.000126	441.833862	-72.739315	864.101135
max	323.299988	0.000252	518.161743	5.090859	1233.212646

	...	SoilMoi40_100cm_inst	SoilTMP0_10cm_inst	SoilTMP100_200cm_inst
count	...	5843.000000	5843.000000	5843.000000
unique	...	NaN	NaN	NaN
top	...	NaN	NaN	NaN
freq	...	NaN	NaN	NaN
mean	...	166.106882	306.142527	301.507732
std	...	56.060541	8.080839	3.100573
min	...	104.426003	293.527222	296.317322
25%	...	117.391998	300.149872	298.836609
50%	...	137.955002	302.815735	301.324585
75%	...	223.673500	312.016769	303.848770
max	...	269.950989	325.580750	308.313293

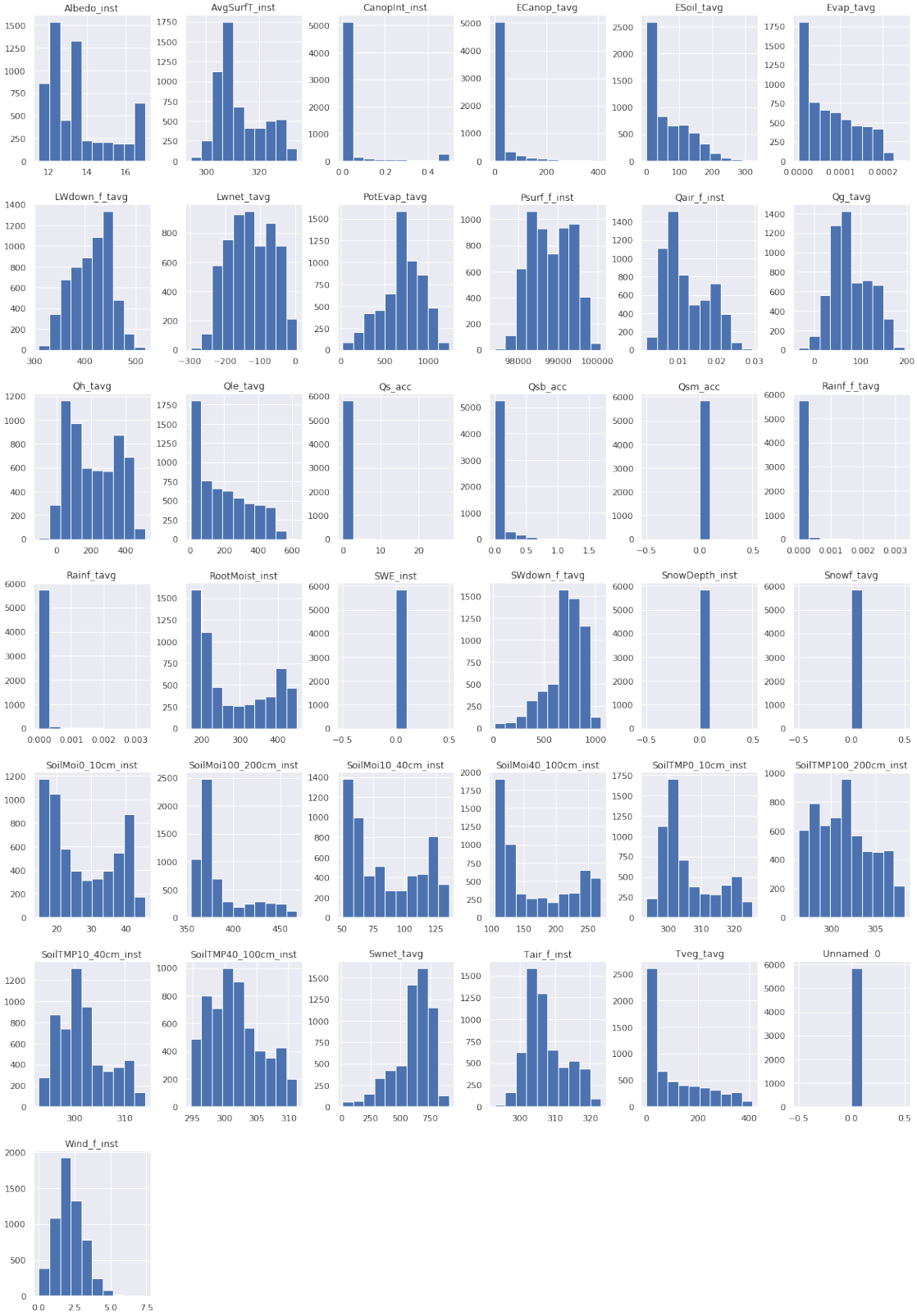
	SoilTMP10_40cm_inst	SoilTMP40_100cm_inst	Swnet_tavg	Tair_f_inst

```
count           5843.000000           5843.000000  5843.000000  5843.000000
unique                  NaN                   NaN          NaN          NaN
top                     NaN                   NaN          NaN          NaN
freq                    NaN                   NaN          NaN          NaN
mean             301.763629            301.737592   607.622805   307.432248
std                4.918302              4.052518   157.954647     5.756286
min              292.833801            294.751099    13.850000   292.790863
25%              297.948441            298.414230   542.605011   303.316788
50%              301.063080            301.309723   639.859985   306.000793
75%              304.532776            304.418961   719.619995   311.177841
max              314.112000            311.308136   909.049988   323.115479

          Tveg_tavg  Wind_f_inst                        geometry
count   5843.000000  5843.000000                            5843
unique          NaN          NaN                            5843
top             NaN          NaN  <map object at 0x7f1b31f9c3c8>
freq            NaN          NaN                               1
mean     107.845247     2.127956                             NaN
std      114.331925     0.958956                             NaN
min        0.000000     0.007467                             NaN
25%       10.980000     1.413756                             NaN
50%       57.799999     2.101675                             NaN
75%      189.979996     2.714424                             NaN
max      412.649994     7.410892                             NaN

[11 rows x 38 columns]
```

In [14]:
```
data.hist(figsize=(20,30))
```

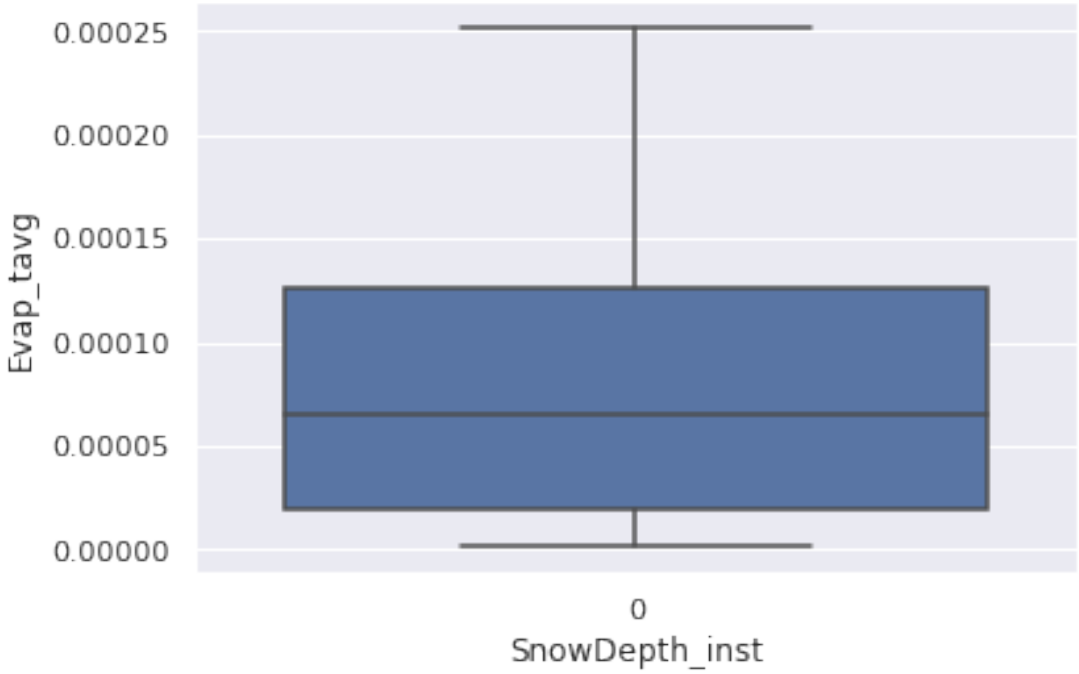
Out[14]:
```
array([[<matplotlib.axes._subplots.AxesSubplot object at 0x7fd6a7e290b8>,
        <matplotlib.axes._subplots.AxesSubplot object at 0x7fd6a79e3550>,
        <matplotlib.axes._subplots.AxesSubplot object at 0x7fd6a7986d68>,
        <matplotlib.axes._subplots.AxesSubplot object at 0x7fd6a79b4710>,
        <matplotlib.axes._subplots.AxesSubplot object at 0x7fd6a795a780>,
        <matplotlib.axes._subplots.AxesSubplot object at 0x7fd6a797edd8>],
       [<matplotlib.axes._subplots.AxesSubplot object at 0x7fd6a792d470>,
        <matplotlib.axes._subplots.AxesSubplot object at 0x7fd6a78d3c88>,
        <matplotlib.axes._subplots.AxesSubplot object at 0x7fd6a78d3cc0>,
        <matplotlib.axes._subplots.AxesSubplot object at 0x7fd6a78a4f60>,
        <matplotlib.axes._subplots.AxesSubplot object at 0x7fd6a7850908>,
        <matplotlib.axes._subplots.AxesSubplot object at 0x7fd6a787e2b0>],
       [<matplotlib.axes._subplots.AxesSubplot object at 0x7fd6a7821c18>,
        <matplotlib.axes._subplots.AxesSubplot object at 0x7fd6a77cf5c0>,
        <matplotlib.axes._subplots.AxesSubplot object at 0x7fd6a77f3f28>,
        <matplotlib.axes._subplots.AxesSubplot object at 0x7fd6a77a08d0>,
        <matplotlib.axes._subplots.AxesSubplot object at 0x7fd6a774f278>,
        <matplotlib.axes._subplots.AxesSubplot object at 0x7fd6a7774be0>],
       [<matplotlib.axes._subplots.AxesSubplot object at 0x7fd6a7721588>,
```

```
  <matplotlib.axes._subplots.AxesSubplot object at 0x7fd6a76c7ef0>,
  <matplotlib.axes._subplots.AxesSubplot object at 0x7fd6a76f4898>,
  <matplotlib.axes._subplots.AxesSubplot object at 0x7fd6a76a1240>,
  <matplotlib.axes._subplots.AxesSubplot object at 0x7fd6a7647ba8>,
  <matplotlib.axes._subplots.AxesSubplot object at 0x7fd6a7675550>],
 [<matplotlib.axes._subplots.AxesSubplot object at 0x7fd6a761aeb8>,
  <matplotlib.axes._subplots.AxesSubplot object at 0x7fd6a75c9860>,
  <matplotlib.axes._subplots.AxesSubplot object at 0x7fd6a75f7208>,
  <matplotlib.axes._subplots.AxesSubplot object at 0x7fd6a759ab70>,
  <matplotlib.axes._subplots.AxesSubplot object at 0x7fd6a7549518>,
  <matplotlib.axes._subplots.AxesSubplot object at 0x7fd6a756de80>],
 [<matplotlib.axes._subplots.AxesSubplot object at 0x7fd6a751c828>,
  <matplotlib.axes._subplots.AxesSubplot object at 0x7fd6a74cb1d0>,
  <matplotlib.axes._subplots.AxesSubplot object at 0x7fd6a74eeb38>,
  <matplotlib.axes._subplots.AxesSubplot object at 0x7fd6a749d4e0>,
  <matplotlib.axes._subplots.AxesSubplot object at 0x7fd6a74c2e48>,
  <matplotlib.axes._subplots.AxesSubplot object at 0x7fd6a746e7f0>],
 [<matplotlib.axes._subplots.AxesSubplot object at 0x7fd6a741e198>,
  <matplotlib.axes._subplots.AxesSubplot object at 0x7fd6a7442b00>,
  <matplotlib.axes._subplots.AxesSubplot object at 0x7fd6a73f14a8>,
  <matplotlib.axes._subplots.AxesSubplot object at 0x7fd6a7396e10>,
  <matplotlib.axes._subplots.AxesSubplot object at 0x7fd6a73c27b8>,
  <matplotlib.axes._subplots.AxesSubplot object at 0x7fd6a7371160>]],
dtype=object)
```


Albedo_inst
AvgSurfT_inst
CanopInt_inst
ECanop_tavg
ESoil_tavg
Evap_tavg
LWdown_f_tavg
Lwnet_tavg
PotEvap_tavg
Psurf_f_inst
Qair_f_inst
Qg_tavg
Qh_tavg
Qle_tavg
Qs_acc
Qsb_acc
Qsm_acc
Rainf_f_tavg
Rainf_tavg
RootMoist_inst
SWE_inst
SWdown_f_tavg
SnowDepth_inst
Snowf_tavg
SoilMoi0_10cm_inst
SoilMoi100_200cm_inst
SoilMoi10_40cm_inst
SoilMoi40_100cm_inst
SoilTMP0_10cm_inst
SoilTMP100_200cm_inst
SoilTMP10_40cm_inst
SoilTMP40_100cm_inst
Swnet_tavg
Tair_f_inst
Tveg_tavg
Unnamed: 0
Wind_f_inst

```
In [15]: sns.boxplot(x="SnowDepth_inst", y="Evap_tavg", data = data)
```

```
Out[15]: <matplotlib.axes._subplots.AxesSubplot at 0x7fd6a60f0f28>
```




```
In [ ]: sns.pairplot(data)
```

```
Out[ ]: <seaborn.axisgrid.PairGrid at 0x7fd6a5fd47b8>
```

```
In [ ]:
```