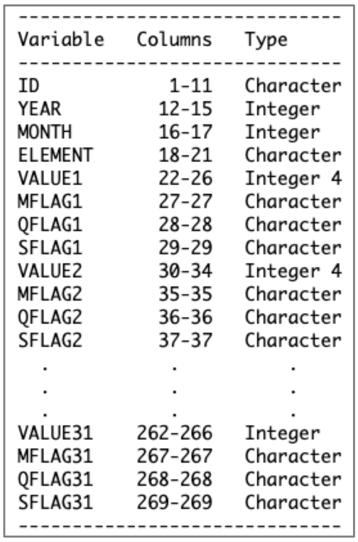
Smarter Transit (Weather of NYC)

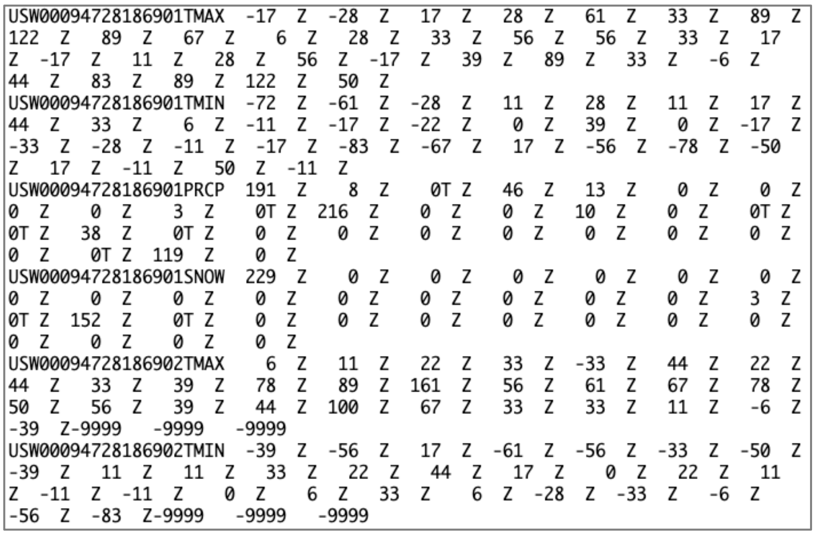
Data profiling, cleaning, and ingestion

An Lee (al7527)

- **Dataset**
  - The weather data comes from NOAA (National Oceanic and Atmospheric Administration). The dataset storage the weather data from 1900s to now by region separately and each distinct "dly" file contains abundant weather conditions, including five core temperature(min & max), snow condition(snowfall & depth) and precipitation, and other 84 elements(wind, soil condition etc.).
  - Each record in a file contains one month of daily data. Please see Figure 1 for the original data's variables and format.



```
------------------------------
Variable   Columns   Type
------------------------------
ID            1-11   Character
YEAR         12-15   Integer
MONTH        16-17   Integer
ELEMENT      18-21   Character
VALUE1       22-26   Integer 4
MFLAG1       27-27   Character
QFLAG1       28-28   Character
SFLAG1       29-29   Character
VALUE2       30-34   Integer 4
MFLAG2       35-35   Character
QFLAG2       36-36   Character
SFLAG2       37-37   Character
  .            .           .
  .            .           .
  .            .           .
VALUE31    262-266   Integer
MFLAG31    267-267   Character
QFLAG31    268-268   Character
SFLAG31    269-269   Character
------------------------------
```



```
USW00094728186901TMAX  -17  Z  -28  Z   17  Z   28  Z   61  Z   33  Z   89  Z
122  Z   89  Z   67  Z    6  Z   28  Z   33  Z   56  Z   56  Z   33  Z   17
Z  -17  Z   11  Z   28  Z   56  Z  -17  Z   39  Z   89  Z   33  Z   -6  Z
44  Z   83  Z   89  Z  122  Z   50  Z
USW00094728186901TMIN  -72  Z  -61  Z  -28  Z   11  Z   28  Z   11  Z   17  Z
44  Z   33  Z    6  Z  -11  Z  -17  Z  -22  Z    0  Z   39  Z    0  Z  -17  Z
-33  Z  -28  Z  -11  Z  -17  Z  -83  Z  -67  Z   17  Z  -56  Z  -78  Z  -50
Z   17  Z  -11  Z   50  Z  -11  Z
USW00094728186901PRCP  191  Z    8  Z    0T Z   46  Z   13  Z    0  Z    0  Z
0  Z    0  Z    3  Z    0T Z  216  Z    0  Z    0  Z   10  Z    0  Z    0T Z
0T Z   38  Z    0T Z    0  Z    0  Z    0  Z    0  Z    0  Z    0  Z    0  Z
0  Z    0T Z  119  Z    0  Z
USW00094728186901SNOW  229  Z    0  Z    0  Z    0  Z    0  Z    0  Z    0  Z
0  Z    0  Z    0  Z    0  Z    0  Z    0  Z    0  Z    0  Z    0  Z    3  Z
0T Z  152  Z    0T Z    0  Z    0  Z    0  Z    0  Z    0  Z    0  Z    0  Z
0  Z    0  Z    0  Z    0  Z
USW00094728186902TMAX    6  Z   11  Z   22  Z   33  Z  -33  Z   44  Z   22  Z
44  Z   33  Z   39  Z   78  Z   89  Z  161  Z   56  Z   61  Z   67  Z   78  Z
50  Z   56  Z   39  Z   44  Z  100  Z   67  Z   33  Z   33  Z   11  Z   -6  Z
-39  Z-9999   -9999   -9999
USW00094728186902TMIN  -39  Z  -56  Z   17  Z  -61  Z  -56  Z  -33  Z  -50  Z
-39  Z   11  Z   11  Z   33  Z   22  Z   44  Z   17  Z    0  Z   22  Z   11
Z  -11  Z  -11  Z    0  Z    6  Z   33  Z    6  Z  -28  Z  -33  Z   -6  Z
-56  Z  -83  Z-9999   -9999   -9999
```

Figure 1: the original data's variables and format

- **Cleaning**

Intending to analyze the correlation between weather of NYC and NYC subway, taxi and crime rate, we need to filter NYC region and pick up those weather factors that might have influence transportation and human activity. Also, we need to let the format be more convenient when user need to filter by time and weather elements.

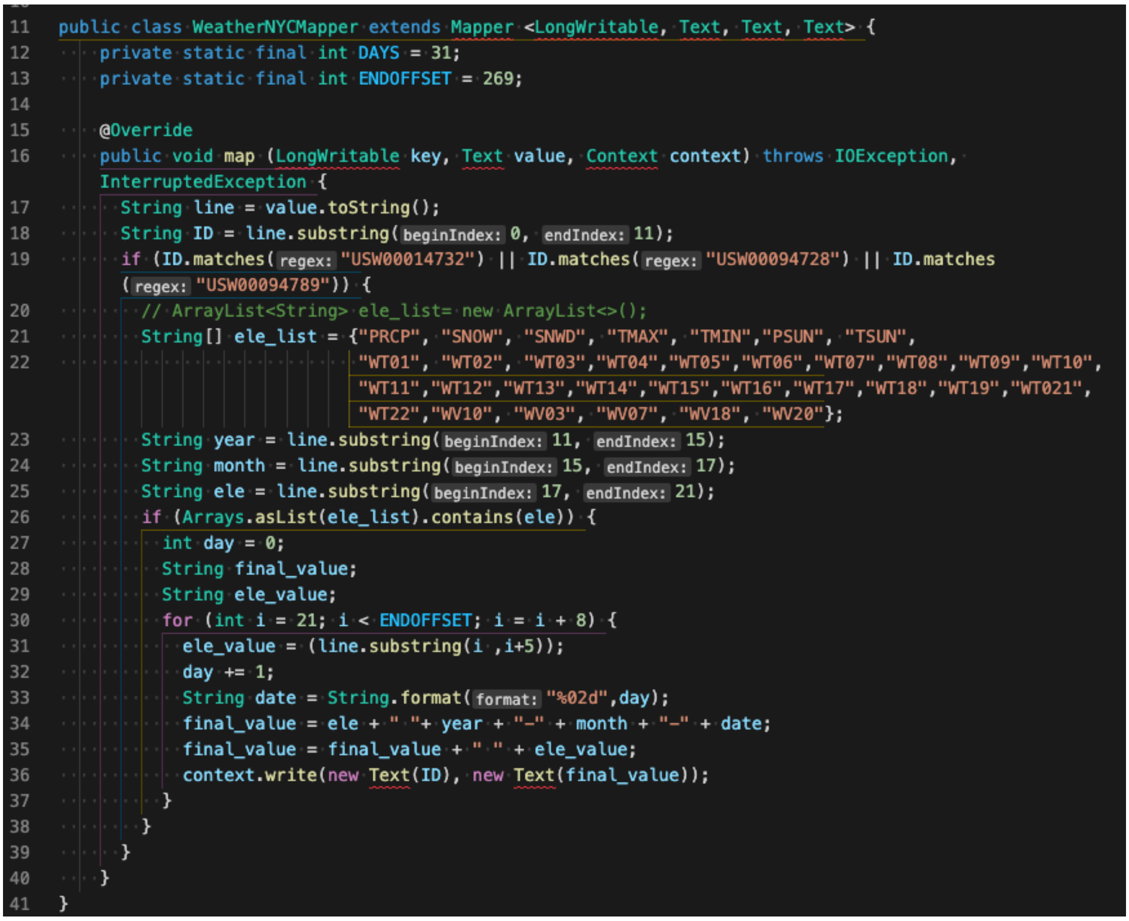
Therefore, we use MapReduce to do the cleaning (See the figure 2 for the source code and the Figure 3 for the elements that we filter out):

Line 18-19 : Identify each file represents which region, and use this region ID to target the data from NYC.

Line 21-26 : Filter out those important weather element that might have influence transportation and human activity.

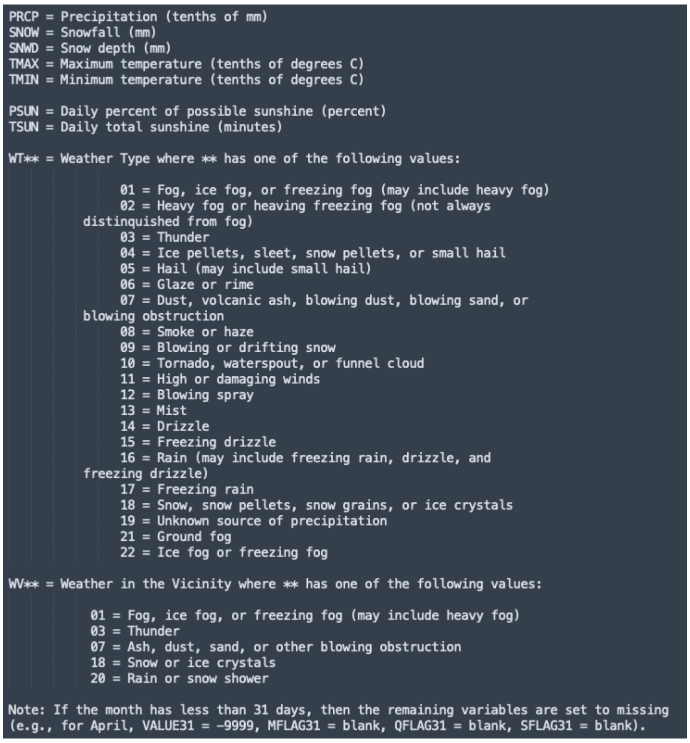
Line 27,33 : Reorganize the "time" format, letting user be more easy to find data by time.

Line 34-35 : Present the specific element and its value.



```
11  public class WeatherNYCMapper extends Mapper <LongWritable, Text, Text, Text> {
12      private static final int DAYS = 31;
13      private static final int ENDOFFSET = 269;
14
15      @Override
16      public void map (LongWritable key, Text value, Context context) throws IOException,
        InterruptedException {
17          String line = value.toString();
18          String ID = line.substring(beginIndex: 0, endIndex: 11);
19          if (ID.matches(regex: "USW00014732") || ID.matches(regex: "USW00094728") || ID.matches
            (regex: "USW00094789")) {
20              // ArrayList<String> ele_list= new ArrayList<>();
21              String[] ele_list = {"PRCP", "SNOW", "SNWD", "TMAX", "TMIN","PSUN", "TSUN",
22                                  "WT01", "WT02", "WT03","WT04","WT05","WT06","WT07","WT08","WT09","WT10",
                                    "WT11","WT12","WT13","WT14","WT15","WT16","WT17","WT18","WT19","WT021",
                                    "WT22","WV10", "WV03", "WV07", "WV18", "WV20"};
23              String year = line.substring(beginIndex: 11, endIndex: 15);
24              String month = line.substring(beginIndex: 15, endIndex: 17);
25              String ele = line.substring(beginIndex: 17, endIndex: 21);
26              if (Arrays.asList(ele_list).contains(ele)) {
27                  int day = 0;
28                  String final_value;
29                  String ele_value;
30                  for (int i = 21; i < ENDOFFSET; i = i + 8) {
31                      ele_value = (line.substring(i ,i+5));
32                      day += 1;
33                      String date = String.format(format: "%02d",day);
34                      final_value = ele + " "+ year + "-" + month + "-" + date;
35                      final_value = final_value + " " + ele_value;
36                      context.write(new Text(ID), new Text(final_value));
37                  }
38              }
39          }
40      }
41  }
```

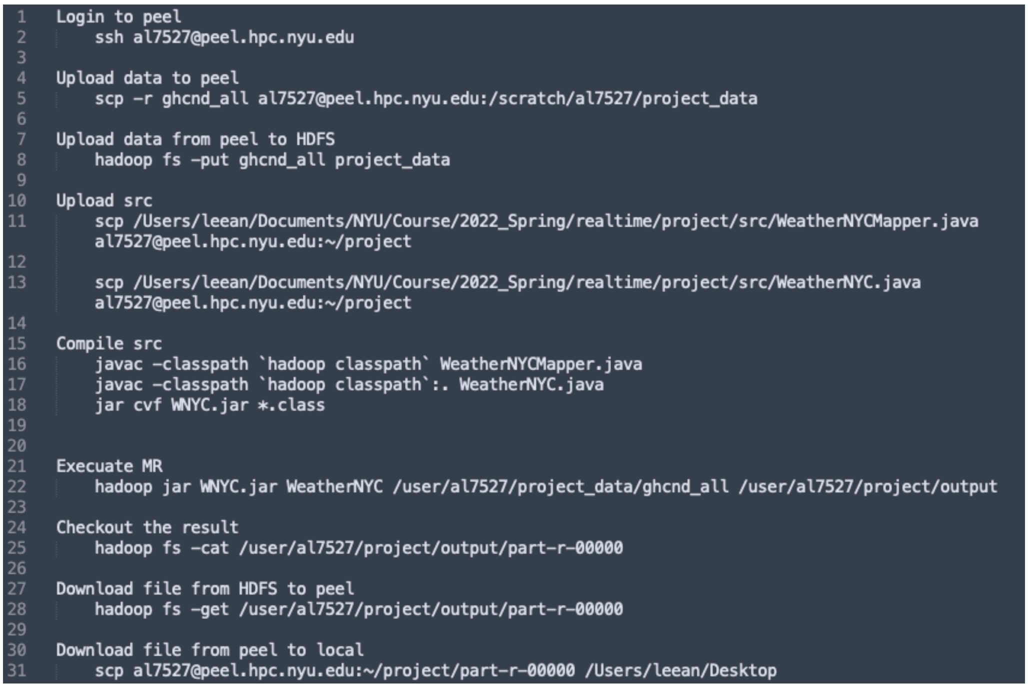
Figure 2: the source code and the elements that we filter out



```
PRCP = Precipitation (tenths of mm)
SNOW = Snowfall (mm)
SNWD = Snow depth (mm)
TMAX = Maximum temperature (tenths of degrees C)
TMIN = Minimum temperature (tenths of degrees C)

PSUN = Daily percent of possible sunshine (percent)
TSUN = Daily total sunshine (minutes)

WT** = Weather Type where ** has one of the following values:

                    01 = Fog, ice fog, or freezing fog (may include heavy fog)
                    02 = Heavy fog or heaving freezing fog (not always
               distinquished from fog)
                    03 = Thunder
                    04 = Ice pellets, sleet, snow pellets, or small hail
                    05 = Hail (may include small hail)
                    06 = Glaze or rime
                    07 = Dust, volcanic ash, blowing dust, blowing sand, or
               blowing obstruction
                    08 = Smoke or haze
                    09 = Blowing or drifting snow
                    10 = Tornado, waterspout, or funnel cloud
                    11 = High or damaging winds
                    12 = Blowing spray
                    13 = Mist
                    14 = Drizzle
                    15 = Freezing drizzle
                    16 = Rain (may include freezing rain, drizzle, and
               freezing drizzle)
                    17 = Freezing rain
                    18 = Snow, snow pellets, snow grains, or ice crystals
                    19 = Unknown source of precipitation
                    21 = Ground fog
                    22 = Ice fog or freezing fog

WV** = Weather in the Vicinity where ** has one of the following values:

                 01 = Fog, ice fog, or freezing fog (may include heavy fog)
                 03 = Thunder
                 07 = Ash, dust, sand, or other blowing obstruction
                 18 = Snow or ice crystals
                 20 = Rain or snow shower

Note: If the month has less than 31 days, then the remaining variables are set to missing
(e.g., for April, VALUE31 = -9999, MFLAG31 = blank, QFLAG31 = blank, SFLAG31 = blank).
```

Figure 3: the elements that we filter out and its meaning

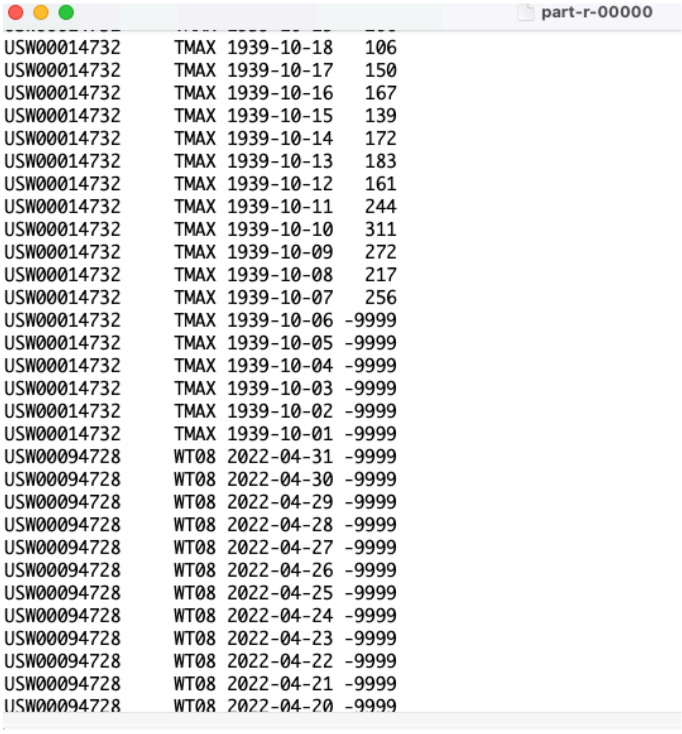
Following command line are for using peel and Hadoop to execute the MapReduce, including steps of upload, compile, execute and download. (See the figure 4)



```
Login to peel
    ssh al7527@peel.hpc.nyu.edu

Upload data to peel
    scp -r ghcnd_all al7527@peel.hpc.nyu.edu:/scratch/al7527/project_data

Upload data from peel to HDFS
    hadoop fs -put ghcnd_all project_data

Upload src
    scp /Users/leean/Documents/NYU/Course/2022_Spring/realtime/project/src/WeatherNYCMapper.java
    al7527@peel.hpc.nyu.edu:~/project

    scp /Users/leean/Documents/NYU/Course/2022_Spring/realtime/project/src/WeatherNYC.java
    al7527@peel.hpc.nyu.edu:~/project

Compile src
    javac -classpath `hadoop classpath` WeatherNYCMapper.java
    javac -classpath `hadoop classpath`:. WeatherNYC.java
    jar cvf WNYC.jar *.class


Execuate MR
    hadoop jar WNYC.jar WeatherNYC /user/al7527/project_data/ghcnd_all /user/al7527/project/output

Checkout the result
    hadoop fs -cat /user/al7527/project/output/part-r-00000

Download file from HDFS to peel
    hadoop fs -get /user/al7527/project/output/part-r-00000

Download file from peel to local
    scp al7527@peel.hpc.nyu.edu:~/project/part-r-00000 /Users/leean/Desktop
```

Figure 4: steps of upload, compile, execute and download when using peel and Hadoop

By doing so, we can get our data more cleaning (see the figure 5)



```
USW00014732     TMAX 1939-10-18   106
USW00014732     TMAX 1939-10-17   150
USW00014732     TMAX 1939-10-16   167
USW00014732     TMAX 1939-10-15   139
USW00014732     TMAX 1939-10-14   172
USW00014732     TMAX 1939-10-13   183
USW00014732     TMAX 1939-10-12   161
USW00014732     TMAX 1939-10-11   244
USW00014732     TMAX 1939-10-10   311
USW00014732     TMAX 1939-10-09   272
USW00014732     TMAX 1939-10-08   217
USW00014732     TMAX 1939-10-07   256
USW00014732     TMAX 1939-10-06 -9999
USW00014732     TMAX 1939-10-05 -9999
USW00014732     TMAX 1939-10-04 -9999
USW00014732     TMAX 1939-10-03 -9999
USW00014732     TMAX 1939-10-02 -9999
USW00014732     TMAX 1939-10-01 -9999
USW00094728     WT08 2022-04-31 -9999
USW00094728     WT08 2022-04-30 -9999
USW00094728     WT08 2022-04-29 -9999
USW00094728     WT08 2022-04-28 -9999
USW00094728     WT08 2022-04-27 -9999
USW00094728     WT08 2022-04-26 -9999
USW00094728     WT08 2022-04-25 -9999
USW00094728     WT08 2022-04-24 -9999
USW00094728     WT08 2022-04-23 -9999
USW00094728     WT08 2022-04-22 -9999
USW00094728     WT08 2022-04-21 -9999
USW00094728     WT08 2022-04-20 -9999
```

Figure 5 : The outcome of cleaning data 's variables and format