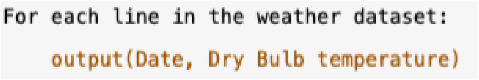
Pseudo Code

I used the same Mapper function for each of the descriptive statistics, as each calculation carried out in the reduce function required the same input from the same two variables; the date and Dry Bulb Temperature. These were the key-value pair inputs for each Reduce function. I decide to create a separate Reduce function for each statistic as I felt it would be easier to trouble shoot the functions locally, prior to implementing them on Hadoop.

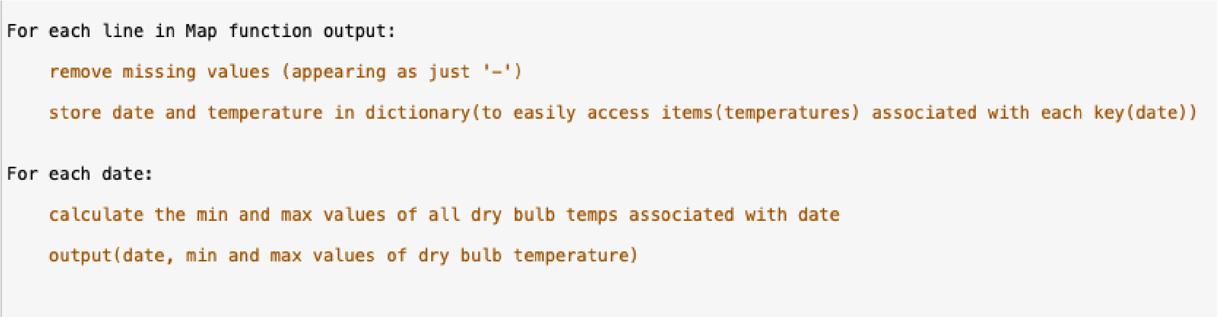
Map Function



```
For each line in the weather dataset:
    output(Date, Dry Bulb temperature)
```

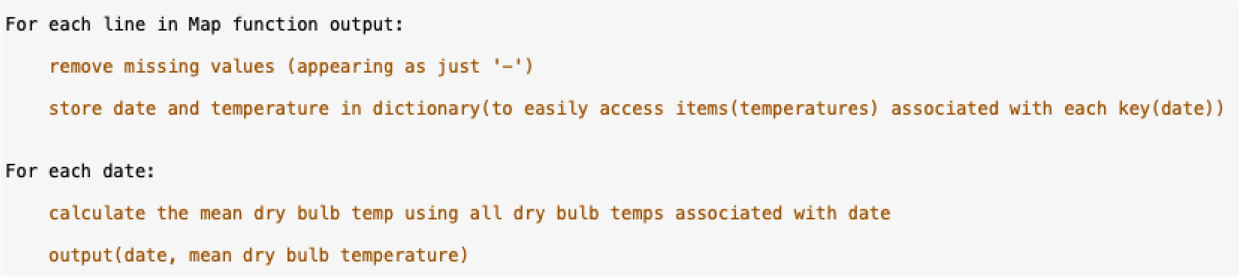
Reduce Functions

1. Finding Daily maximum and minimum Dry Bulb Temp across all weather station



```
For each line in Map function output:
    remove missing values (appearing as just '-')
    store date and temperature in dictionary(to easily access items(temperatures) associated with each key(date))

For each date:
    calculate the min and max values of all dry bulb temps associated with date
    output(date, min and max values of dry bulb temperature)
```

2. Finding Daily mean of Dry Bulb Temperature over all weather stations



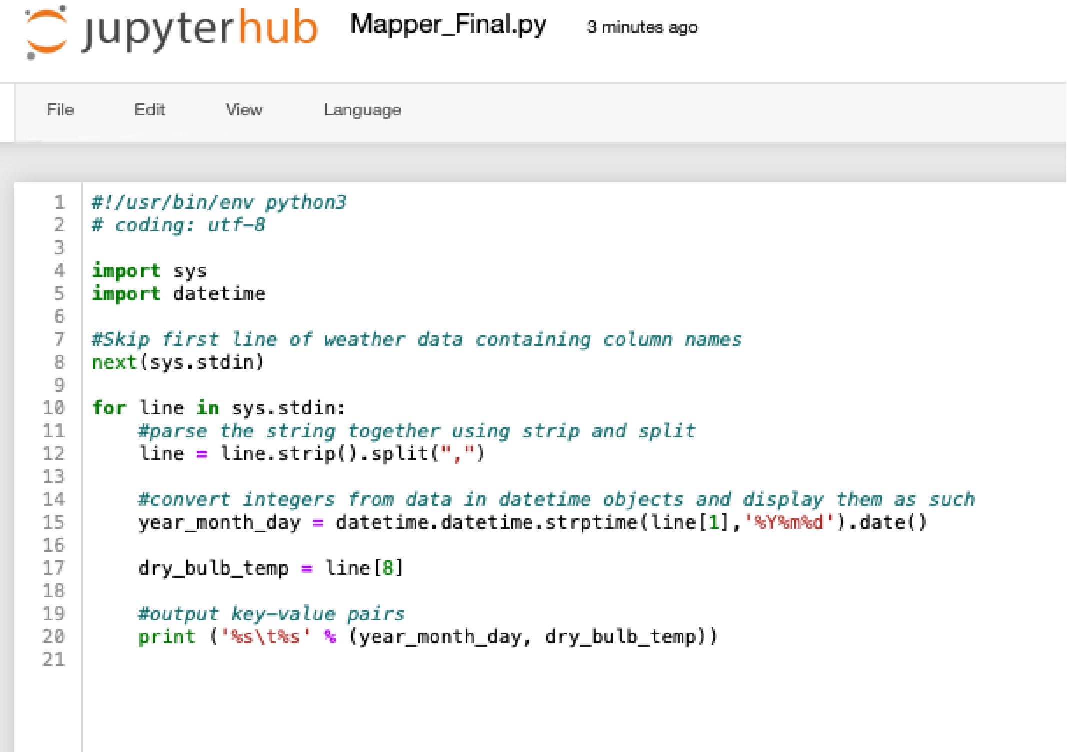
```
For each line in Map function output:
    remove missing values (appearing as just '-')
    store date and temperature in dictionary(to easily access items(temperatures) associated with each key(date))

For each date:
    calculate the mean dry bulb temp using all dry bulb temps associated with date
    output(date, mean dry bulb temperature)
```

3. Finding Daily standard deviation of Dry Bulb Temperature over all weather stations

```
For each line in Map function output:
    remove missing values (appearing as just '-')
    store date and temperature in dictionary(to easily access items(temperatures) associated with each key(date))

For each date:
    calculate the standard deviation using all dry bulb temps associated with date
    output(date, stdev of dry bulb temperature)
```

Mapper and Reduce Python Scripts

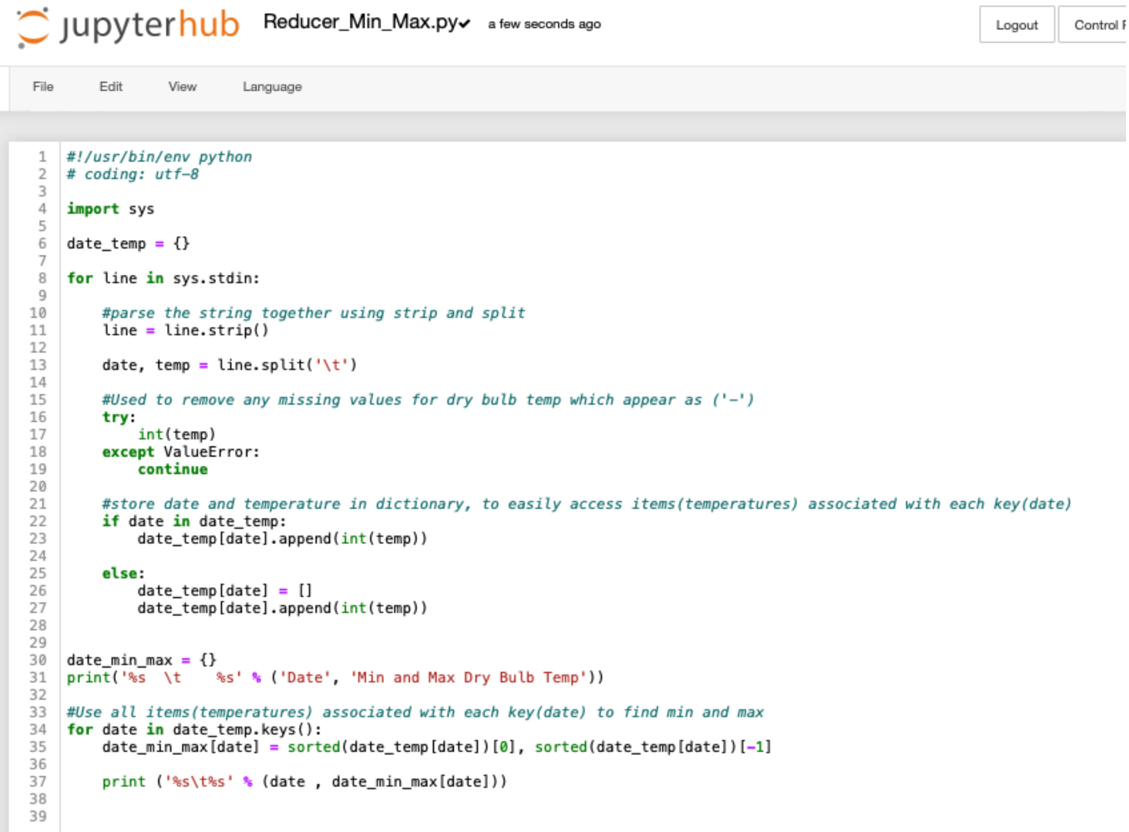
Mapper Script



```python
#!/usr/bin/env python3
# coding: utf-8

import sys
import datetime

#Skip first line of weather data containing column names
next(sys.stdin)

for line in sys.stdin:
    #parse the string together using strip and split
    line = line.strip().split(",")

    #convert integers from data in datetime objects and display them as such
    year_month_day = datetime.datetime.strptime(line[1],'%Y%m%d').date()

    dry_bulb_temp = line[8]

    #output key-value pairs
    print ('%s\t%s' % (year_month_day, dry_bulb_temp))

```

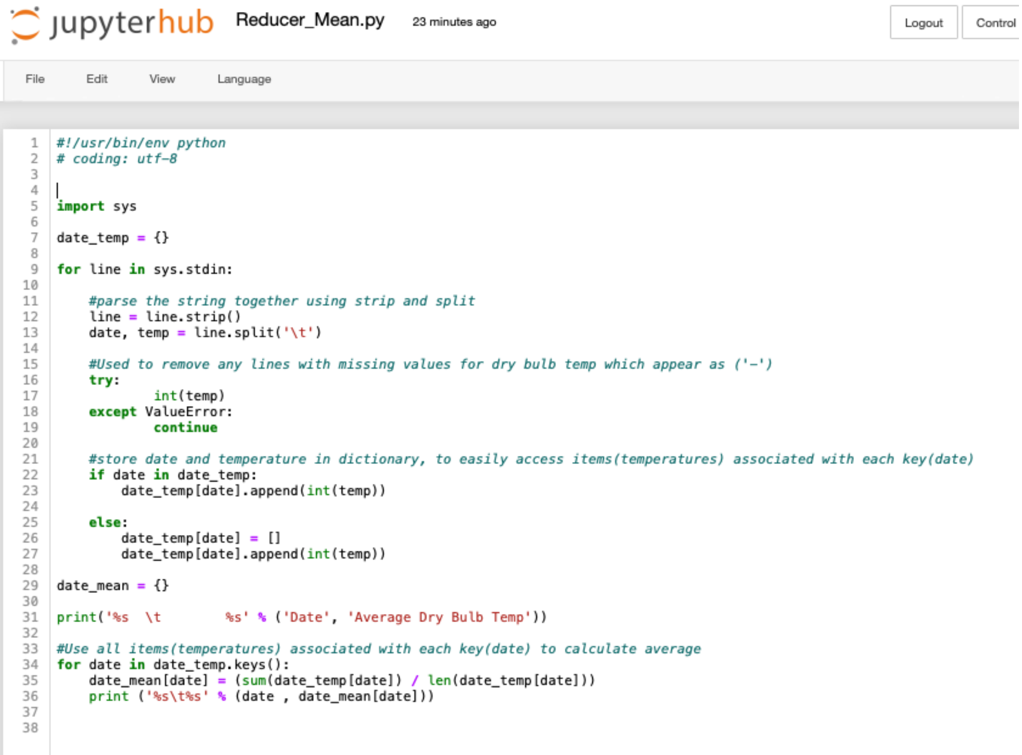
Reduce Scripts

1. Finding Daily maximum and minimum Dry Bulb Temp across all weather station



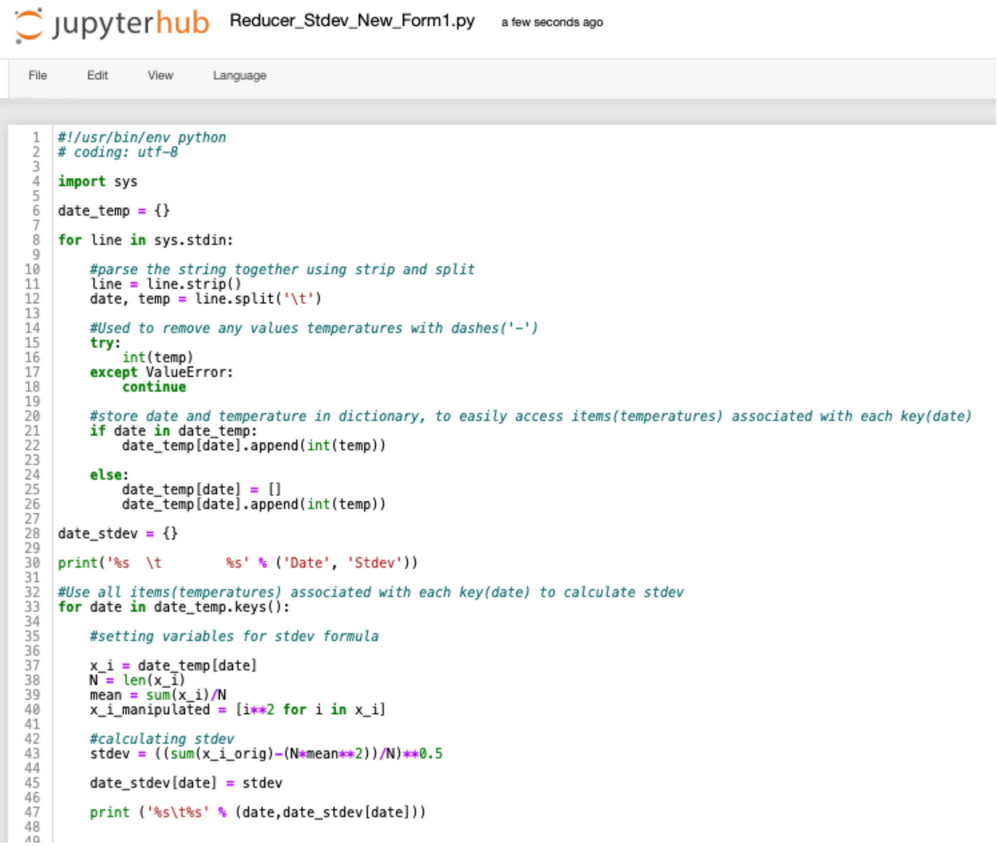
```python
#!/usr/bin/env python
# coding: utf-8

import sys

date_temp = {}

for line in sys.stdin:

    #parse the string together using strip and split
    line = line.strip()

    date, temp = line.split('\t')

    #Used to remove any missing values for dry bulb temp which appear as ('-')
    try:
        int(temp)
    except ValueError:
        continue

    #store date and temperature in dictionary, to easily access items(temperatures) associated with each key(date)
    if date in date_temp:
        date_temp[date].append(int(temp))

    else:
        date_temp[date] = []
        date_temp[date].append(int(temp))


date_min_max = {}
print('%s  \t    %s' % ('Date', 'Min and Max Dry Bulb Temp'))

#Use all items(temperatures) associated with each key(date) to find min and max
for date in date_temp.keys():
    date_min_max[date] = sorted(date_temp[date])[0], sorted(date_temp[date])[-1]

    print ('%s\t%s' % (date , date_min_max[date]))
```

2. Finding Daily mean of Dry Bulb Temperature over all weather stations



```python
#!/usr/bin/env python
# coding: utf-8


import sys

date_temp = {}

for line in sys.stdin:

    #parse the string together using strip and split
    line = line.strip()
    date, temp = line.split('\t')

    #Used to remove any lines with missing values for dry bulb temp which appear as ('-')
    try:
            int(temp)
    except ValueError:
            continue

    #store date and temperature in dictionary, to easily access items(temperatures) associated with each key(date)
    if date in date_temp:
        date_temp[date].append(int(temp))

    else:
        date_temp[date] = []
        date_temp[date].append(int(temp))

date_mean = {}

print('%s  \t        %s' % ('Date', 'Average Dry Bulb Temp'))

#Use all items(temperatures) associated with each key(date) to calculate average
for date in date_temp.keys():
    date_mean[date] = (sum(date_temp[date]) / len(date_temp[date]))
    print ('%s\t%s' % (date , date_mean[date]))
```

3. Finding Daily standard deviation of Dry Bulb Temperature over all weather stations

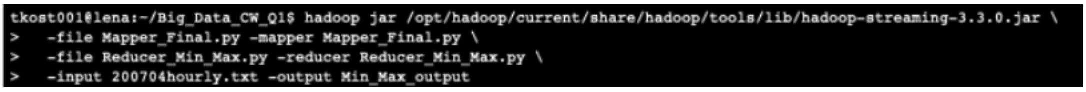


```python
#!/usr/bin/env python
# coding: utf-8

import sys

date_temp = {}

for line in sys.stdin:

    #parse the string together using strip and split
    line = line.strip()
    date, temp = line.split('\t')

    #Used to remove any values temperatures with dashes('-')
    try:
        int(temp)
    except ValueError:
        continue

    #store date and temperature in dictionary, to easily access items(temperatures) associated with each key(date)
    if date in date_temp:
        date_temp[date].append(int(temp))

    else:
        date_temp[date] = []
        date_temp[date].append(int(temp))

date_stdev = {}

print('%s  \t        %s' % ('Date', 'Stdev'))

#Use all items(temperatures) associated with each key(date) to calculate stdev
for date in date_temp.keys():

    #setting variables for stdev formula

    x_i = date_temp[date]
    N = len(x_i)
    mean = sum(x_i)/N
    x_i_manipulated = [i**2 for i in x_i]

    #calculating stdev
    stdev = ((sum(x_i_orig)-(N*mean**2))/N)**0.5

    date_stdev[date] = stdev

    print ('%s\t%s' % (date,date_stdev[date]))
```

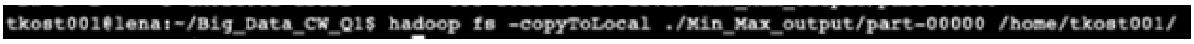
Running MapReduce program on Hadoop and Corresponding Outputs

1. Finding Daily maximum and minimum Dry Bulb Temp across all weather station

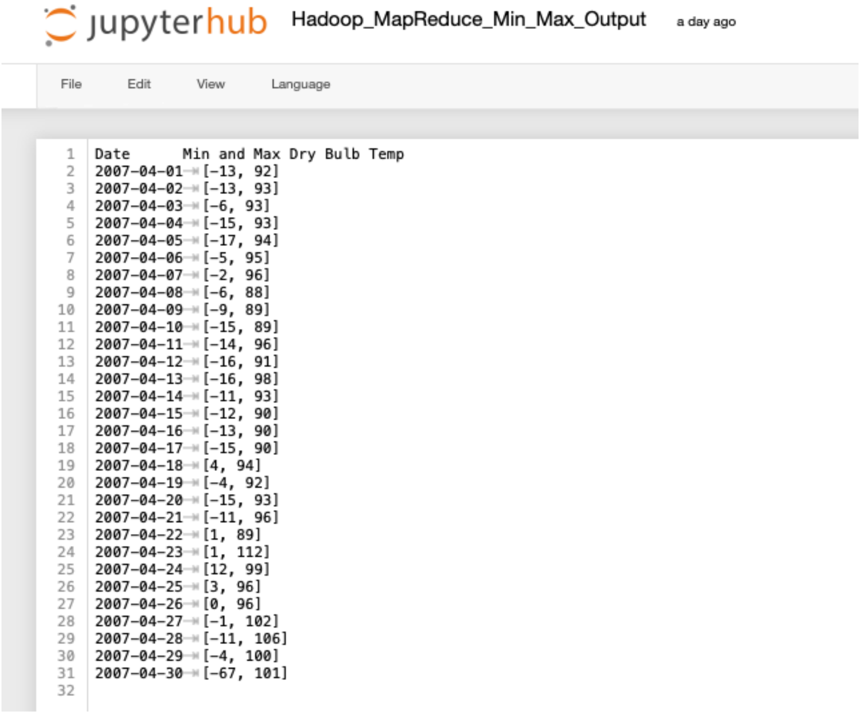
The following Hadoop commands were used to run the program and copy the output from the HDFS to the local file system. The output for each task is displayed below the two screenshots of the commands.



```
tkost001@lena:~/Big_Data_CW_Q1$ hadoop jar /opt/hadoop/current/share/hadoop/tools/lib/hadoop-streaming-3.3.0.jar \
>   -file Mapper_Final.py -mapper Mapper_Final.py \
>   -file Reducer_Min_Max.py -reducer Reducer_Min_Max.py \
>   -input 200704hourly.txt -output Min_Max_output
```

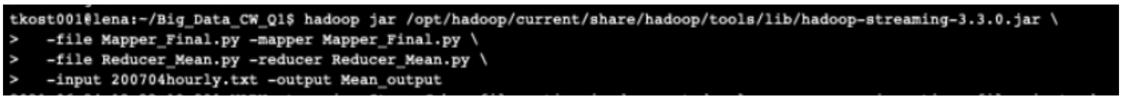


```
tkost001@lena:~/Big_Data_CW_Q1$ hadoop fs -copyToLocal ./Min_Max_output/part-00000 /home/tkost001/
```

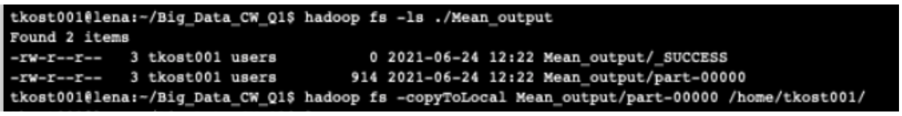


```
Date	Min and Max Dry Bulb Temp
2007-04-01	[-13, 92]
2007-04-02	[-13, 93]
2007-04-03	[-6, 93]
2007-04-04	[-15, 93]
2007-04-05	[-17, 94]
2007-04-06	[-5, 95]
2007-04-07	[-2, 96]
2007-04-08	[-6, 88]
2007-04-09	[-9, 89]
2007-04-10	[-15, 89]
2007-04-11	[-14, 96]
2007-04-12	[-16, 91]
2007-04-13	[-16, 98]
2007-04-14	[-11, 93]
2007-04-15	[-12, 90]
2007-04-16	[-13, 90]
2007-04-17	[-15, 90]
2007-04-18	[4, 94]
2007-04-19	[-4, 92]
2007-04-20	[-15, 93]
2007-04-21	[-11, 96]
2007-04-22	[1, 89]
2007-04-23	[1, 112]
2007-04-24	[12, 99]
2007-04-25	[3, 96]
2007-04-26	[0, 96]
2007-04-27	[-1, 102]
2007-04-28	[-11, 106]
2007-04-29	[-4, 100]
2007-04-30	[-67, 101]
```

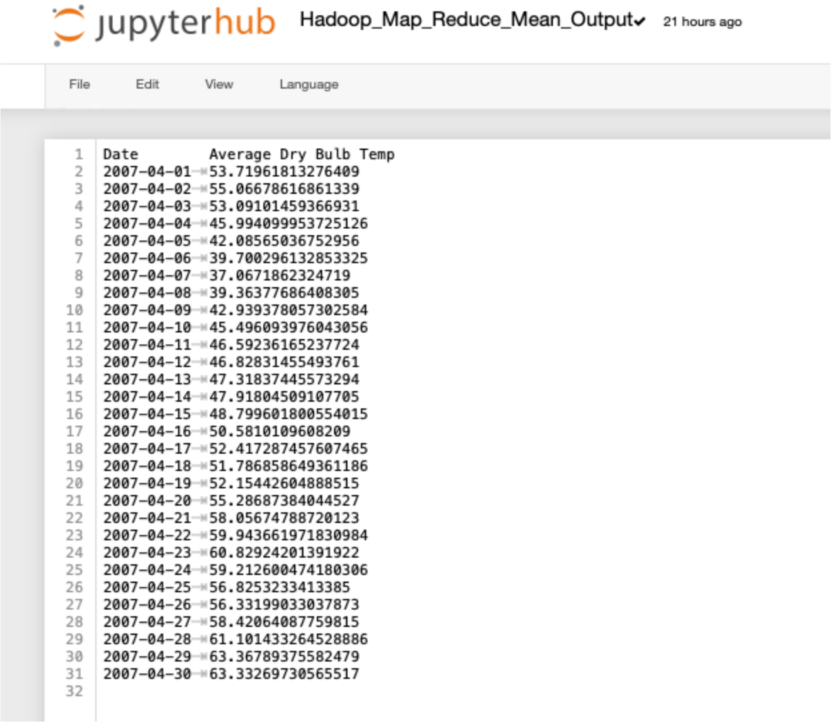
2. Finding Daily mean of Dry Bulb Temperature over all weather stations



```
tkost001@lena:~/Big_Data_CW_Q1$ hadoop jar /opt/hadoop/current/share/hadoop/tools/lib/hadoop-streaming-3.3.0.jar \
>   -file Mapper_Final.py -mapper Mapper_Final.py \
>   -file Reducer_Mean.py -reducer Reducer_Mean.py \
>   -input 200704hourly.txt -output Mean_output
```

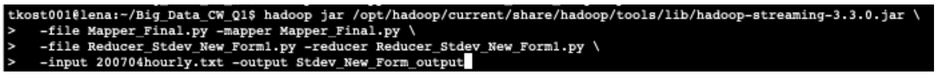


```
tkost001@lena:~/Big_Data_CW_Q1$ hadoop fs -ls ./Mean_output
Found 2 items
-rw-r--r--   3 tkost001 users          0 2021-06-24 12:22 Mean_output/_SUCCESS
-rw-r--r--   3 tkost001 users        914 2021-06-24 12:22 Mean_output/part-00000
tkost001@lena:~/Big_Data_CW_Q1$ hadoop fs -copyToLocal Mean_output/part-00000 /home/tkost001/
```

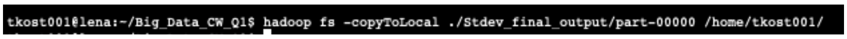


```
Date		Average Dry Bulb Temp
2007-04-01	53.71961813276409
2007-04-02	55.06678616861339
2007-04-03	53.09101459366931
2007-04-04	45.994099953725126
2007-04-05	42.08565036752956
2007-04-06	39.700296132853325
2007-04-07	37.0671862324719
2007-04-08	39.36377686408305
2007-04-09	42.939378057302584
2007-04-10	45.496093976043056
2007-04-11	46.59236165237724
2007-04-12	46.82831455493761
2007-04-13	47.31837445573294
2007-04-14	47.91804509107705
2007-04-15	48.799601800554015
2007-04-16	50.5810109608209
2007-04-17	52.417287457607465
2007-04-18	51.786858649361186
2007-04-19	52.15442604888515
2007-04-20	55.28687384044527
2007-04-21	58.05674788720123
2007-04-22	59.943661971830984
2007-04-23	60.82924201391922
2007-04-24	59.212600474180306
2007-04-25	56.8253233413385
2007-04-26	56.33199033037873
2007-04-27	58.42064087759815
2007-04-28	61.101433264528886
2007-04-29	63.36789375582479
2007-04-30	63.33269730565517
```

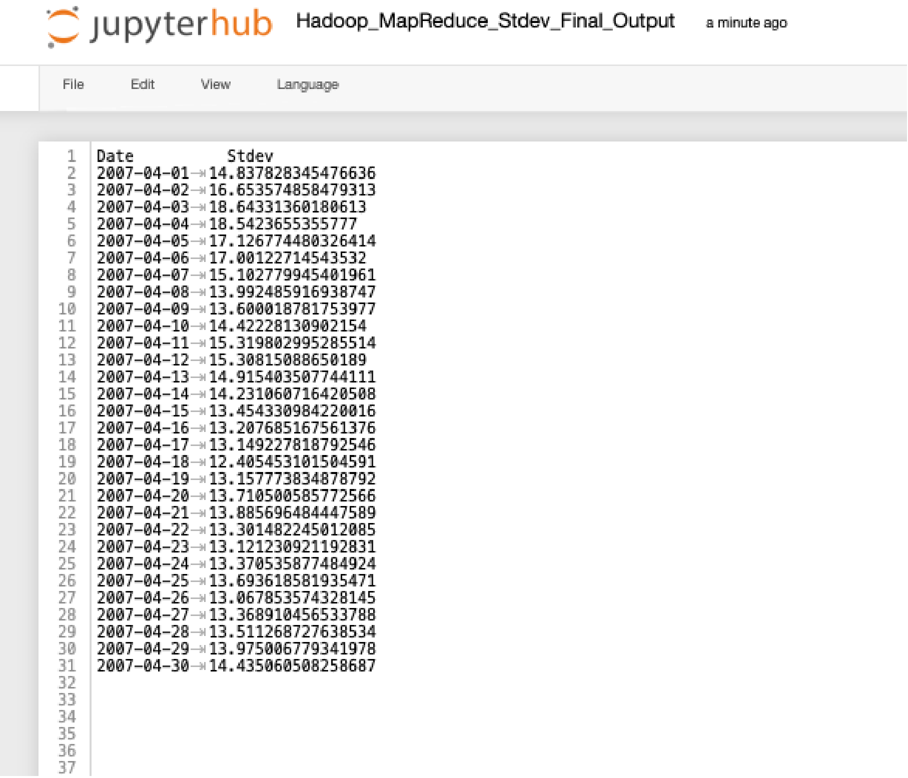
3. Finding Daily standard deviation of Dry Bulb Temperature over all weather stations



```
tkost001@lena:~/Big_Data_CW_Q1$ hadoop jar /opt/hadoop/current/share/hadoop/tools/lib/hadoop-streaming-3.3.0.jar \
>   -file Mapper_Final.py -mapper Mapper_Final.py \
>   -file Reducer_Stdev_New_Form1.py -reducer Reducer_Stdev_New_Form1.py \
>   -input 200704hourly.txt -output Stdev_New_Form_output
```



```
tkost001@lena:~/Big_Data_CW_Q1$ hadoop fs -copyToLocal ./Stdev_final_output/part-00000 /home/tkost001/
```



```
Date            Stdev
2007-04-01	14.837828345476636
2007-04-02	16.653574858479313
2007-04-03	18.64331360180613
2007-04-04	18.5423655355777
2007-04-05	17.126774480326414
2007-04-06	17.00122714543532
2007-04-07	15.102779945401961
2007-04-08	13.992485916938747
2007-04-09	13.600018781753977
2007-04-10	14.42228130902154
2007-04-11	15.319802995285514
2007-04-12	15.30815088650189
2007-04-13	14.915403507744111
2007-04-14	14.231060716420508
2007-04-15	13.454330984220016
2007-04-16	13.207685167561376
2007-04-17	13.149227818792546
2007-04-18	12.405453101504591
2007-04-19	13.157773834878792
2007-04-20	13.710500585772566
2007-04-21	13.885696484447589
2007-04-22	13.301482245012085
2007-04-23	13.121230921192831
2007-04-24	13.370535877484924
2007-04-25	13.693618581935471
2007-04-26	13.067853574328145
2007-04-27	13.368910456533788
2007-04-28	13.511268727638534
2007-04-29	13.975006779341978
2007-04-30	14.435060508258687
```