CLOUD COMPUTING PROJECT 2

BIG DATA ANALYSIS – WEATHER

BY

ALEKHYA MALYALA

M12981264

Analysis

Input: 19 years (2000-2019) weather dataset for all the stations starting with US and the elements are TMAX and TMIN

In the dataset, the temperatures are represented in tenths of degree Celsius which is different from actual Celsius value.

Example: If the tenths of degree Celsius value is 522 then the actual Celsius value is 522/10 i.e. 52.2

a) AVG TMAX, TMIN for each year excluding abnormalities or missing data

Output:

Year	Measurement	Avg.Value (tenths of °C)
2000	TMAX	175.584256706
2000	TMIN	44.3082903296
2001	TMAX	178.747558648
2001	TMIN	47.9238293819
2002	TMAX	177.229975514
2002	TMIN	46.5490911032
2003	TMAX	177.099600803
2003	TMIN	48.6285186518
2004	TMAX	174.506802106
2004	TMIN	49.0952521104
2005	TMAX	177.660982462
2005	TMIN	49.8054192201
2006	TMAX	180.773615979
2006	TMIN	50.8548743305
2007	TMAX	178.219864159
2007	TMIN	49.0736120122
2008	TMAX	170.09655467
2008	TMIN	40.7394844915

2009	TMAX	168.693988733
2009	TMIN	43.3658121143
2010	TMAX	171.51778809
2010	TMIN	47.2613962352
2011	TMAX	172.46933642
2011	TMIN	46.0545355887
2012	TMAX	183.914808379
2012	TMIN	53.4819231631
2013	TMAX	167.31434867
2013	TMIN	43.0010399546
2014	TMAX	168.439429894
2014	TMIN	43.8491942761
2015	TMAX	177.23014761
2015	TMIN	53.5361682807
2016	TMAX	179.074322175
2016	TMIN	55.1074627805
2017	TMAX	176.64585266
2017	TMIN	52.8118997978
2018	TMAX	171.321408178
2018	TMIN	49.4530738941
2019	TMAX	73.0315200726
2019	TMIN	-36.5548757831

b) MAX TMAX, TMIN for each year excluding abnormalities or missing data

For MAX TMAX per year:

Output:

Year	Measurement	Max Value (tenths of °C)
2000	TMAX	522
2001	TMAX	528
2002	TMAX	533
2003	TMAX	533
2004	TMAX	517
2005	TMAX	539
2006	TMAX	528
2007	TMAX	539
2008	TMAX	528
2009	TMAX	533
2010	TMAX	517
2011	TMAX	511
2012	TMAX	537
2013	TMAX	539
2014	TMAX	522
2015	TMAX	556
2016	TMAX	539
2017	TMAX	528
2018	TMAX	528
2019	TMAX	417

For MIN TMIN per year:

Output:

Year	Measurement	Min Value (tenths of °C)
2000	TMIN	-578
2001	TMIN	-528
2002	TMIN	-472
2003	TMIN	-500
2004	TMIN	-533
2005	TMIN	-550
2006	TMIN	-528
2007	TMIN	-539
2008	TMIN	-578
2009	TMIN	-556
2010	TMIN	-533
2011	TMIN	-517
2012	TMIN	-544
2013	TMIN	-528
2014	TMIN	-500
2015	TMIN	-528
2016	TMIN	-469
2017	TMIN	-520
2018	TMIN	-478
2019	TMIN	-494

c) 5 hottest , 5 coldest weather stations for each year excluding abnormalities or missing data

For 5 hottest weather stations per year:

Output:

Year	Station_ID	Max Temp. (tenths of °C)
2000	USC00042319	522
2000	USC00042319	511
2000	USC00042319	506
2000	USC00042319	506
2000	USC00042319	506
2001	USC00042319	528
2001	USC00042319	528
2001	USC00042319	517
2001	USC00042319	511
2001	USC00042319	511
2002	USC00042319	533
2002	USC00042319	528
2002	USC00042319	528
2002	USC00042319	528
2002	USC00042319	517
2003	USC00042319	533
2003	USR0000CMEA	528
2003	USC00042319	522
2003	USC00042319	522
2003	USR0000AHAV	522
2004	USC00042319	517
2004	USC00042319	517
2004	USC00024761	511
2004	USC00024761	506
2004	USC00042319	506
2005	USC00042319	539
2005	USC00042319	533
2005	USC00042319	533
2005	USC00042319	528
2005	USC00042319	528
2006	USC00042319	528
2006	USC00042319	522
2006	USC00042319	517
2006	USC00042319	517
2006	USC00042319	517
2007	USC00042319	539
2007	USC00042319	528
2007	USW00053139	523

2007	USC00042319	522
2007	USC00042319	522
2008	USC00044297	528
2008	USC00042319	528
2008	USC00042319	522
2008	USC00024761	511
2008	USC00044297	511
2009	USC00042319	533
2009	USC00042319	522
2009	USC00042319	522
2009	USC00042319	517
2009	USC00042319	517
2010	USC00042319	517
2010	USC00042319	517
2010	USC00042319	517
2010	USC00042319	517
2010	USR0000AHAV	511
2011	USC00042319	511
2011	USC00042319	511
2011	USC00042319	506
2011	USC00042319	506
2011	USC00042319	500
2012	USS0005N23S	537
2012	USC00042319	533
2012	USC00042319	522
2012	USC00042319	517
2012	USC00042319	517
2013	USC00042319	539
2013	USW00004134	533
2013	USC00042319	533
2013	USC00042319	533
2013	USC00044297	528
2014	USC00042319	522
2014	USC00042319	517
2014	USW00053139	511
2014	USC00042319	506
2014	USC00042319	506
2015	USR0000HKAU	556
2015	USR0000HKAU	556
2015	USR0000HKAU	539
2015	USC00042319	517
2015	USC00042319	511
2016	USR0000CBEV	539
2016	USC00042319	528
2016	USC00042319	522
2016	USC00040924	522

2016	USC00042319	522
2017	USC00042319	528
2017	USC00042319	528
2017	USC00042319	528
2017	USC00042319	528
2017	USC00021050	522
2018	USC00042319	528
2018	USC00042319	528
2018	USC00042319	528
2018	USC00042319	528
2018	USC00042319	522
2019	USW00022010	417
2019	USC00415048	411
2019	USW00012907	406
2019	USC00417624	406
2019	USR0000TFAL	400

For 5 coldest weather stations per year:

Output:

Year	**Station_ID**	**Min Temp. (tenths of °C)**
2000	USC00501684	-578
2000	USC00505644	-578
2000	USC00505644	-550
2000	USC00501684	-550
2000	USC00508140	-539
2001	USW00026508	-528
2001	USR0000ABCA	-511
2001	USW00026508	-506
2001	USS0051R01S	-480
2001	USW00026508	-478
2002	USR0000AKAI	-472
2002	USS0051R01S	-470
2002	USS0050S01S	-470
2002	USR0000ABEV	-467
2002	USC00503212	-461
2003	USC00501492	-500
2003	USS0051R01S	-494
2003	USC00501492	-490
2003	USW00026533	-483
2003	USC00509869	-480
2004	USC00501684	-533
2004	USC00501684	-522
2004	USC00502568	-506

2004	USS0045O10S	-500
2004	USS0045O10S	-500
2005	USC00501684	-550
2005	USC00509313	-528
2005	USC00501684	-522
2005	USC00501684	-522
2005	USC00501684	-522
2006	USR0000ASEL	-528
2006	USC00501492	-522
2006	USC00501492	-517
2006	USC00501492	-517
2006	USC00501492	-511
2007	USC00501684	-539
2007	USC00501684	-511
2007	USC00501684	-506
2007	USS0045R01S	-501
2007	USC00501684	-500
2008	USC00501684	-578
2008	USC00501684	-578
2008	USC00501684	-561
2008	USC00501684	-539
2008	USC00501684	-533
2009	USC00501684	-556
2009	USC00502101	-556
2009	USC00502101	-550
2009	USC00501684	-544
2009	USC00502101	-544
2010	USC00501684	-533
2010	USC00502101	-533
2010	USC00501684	-528
2010	USC00502101	-528
2010	USS0051R01S	-519
2011	USC00509869	-517
2011	USS0045R01S	-507
2011	USS0045R01S	-503
2011	USS0051R01S	-502
2011	USS0051R01S	-502
2012	USC00503165	-544
2012	USC00503165	-544
2012	USC00503165	-539
2012	USC00503212	-539
2012	USS0051R01S	-536
2013	USC00502339	-528

2013	USC00501684	-522
2013	USC00501684	-511
2013	USC00501684	-511
2013	USC00502339	-511
2014	USC00501684	-500
2014	USC00501684	-494
2014	USC00501684	-489
2014	USC00501684	-483
2014	USC00501684	-483
2015	USC00502339	-528
2015	USC00502339	-511
2015	USC00501684	-506
2015	USC00501684	-506
2015	USC00502339	-506
2016	USS0051R01S	-469
2016	USR0000ACHL	-467
2016	USC00501684	-467
2016	USR0000ACHL	-461
2016	USR0000ACHL	-461
2017	USS0051R01S	-520
2017	USR0000ASLC	-506
2017	USW00026529	-506
2017	USW00026529	-506
2017	USS0051R01S	-503
2018	USC00501684	-478
2018	USC00501684	-478
2018	USR0000ANOR	-478
2018	USR0000AKAV	-478
2018	USW00096406	-475
2019	USC00509891	-494
2019	USC00501684	-489
2019	USC00211840	-489
2019	USC00218618	-489
2019	USC00211840	-489

d) Hottest and coldest day and corresponding weather stations in the entire dataset

For Hottest day:

Output:

Date_id	**Station**	**Value**
20150213	USR0000HKAU	556

For Coldest day:

Output:

Date_id	Station	Value
20000101	USC00501684	-578

Map Reduce Usage:

In this project, I used MapReduce to perform data analysis for the given tasks. I preferred to use MapReduce because coding in MapReduce is simpler to me when compared to other techniques. Also, I'm familiar with it because I did the previous assignment using MapReduce. Furthermore it requires less RAM as the containers used by the process will be freed once the job is done so that the containers can be used again for another job execution.

Appendix

//mapper.py

```
#!/usr/bin/env python
import sys
def mapToColDict(lst):
    return {'ID':lst[0],
            'DATE':lst[1],
            'TYPE':lst[2],
            'VALUE':lst[3],
            'MFLAG':lst[4],
            'QFLAG':lst[5],
            'SFLAG':lst[6],
            'OBSTIME':lst[7]}

for line in sys.stdin:
    parse = line.strip().upper().split(',')
    row = mapToColDict(parse)
    if 'TMAX' != row['TYPE'] and 'TMIN' != row['TYPE']:
        continue
    if row['VALUE'] == -9999:
        continue
    if row['SFLAG'] == '':
        continue
    if row['QFLAG'] != '':
        continue
    if row['MFLAG'] == 'P':
        contiune
```

```
    print '%s,%s,%s,%s,' % (row['DATE'],row['ID'],row['TYPE'],row['VALUE'])
//reducer.py
#!/usr/bin/env python
import sys
import operator
current_year = None
max_count = 0
min_count = 0
avg_max = 0
avg_min = 0
hottest = []
coldest = []
max = (-9999,",")
min = (99999,",")

for line in sys.stdin:
    line = line.strip().split(',')

    date = line[0]
    year = date[:4]
    id = line[1]
    metric = line[2]
    value = line[3]
    try:
        value = int(value)
    except ValueError:
        continue
    if current_year is None:
```

```
    current_year = year

if current_year != year:
    print 'Year: %s' % current_year
    print 'Average TMAX: %s' % (avg_max * 1.0 / max_count)
    print 'Average TMIN: %s' % (avg_min * 1.0 / min_count)
    print 'Max TMAX: %s' % (hottest[0][0])
    print 'Min TMIN: %s' % (coldest[0][0])
    print 'Hottest Stations %s' % ([x[1] for x in hottest])
    print 'Hottest Station Values %s' % ([x[0] for x in hottest])
    print 'Coldest Stations %s' % ([x[1] for x in coldest])
    print 'Coldest Station Values %s' % ([x[0] for x in coldest])
    print '--------------------------------------------------'

    current_year = year
    max_count = 0
    min_count = 0
    avg_max = 0
    avg_min = 0
    hottest = []
    coldest = []

if metric == 'TMAX':
    avg_max += value
    max_count += 1
    if max[0] < value:
        max = (value,id,date)
    hottest.append((value,id,date))
```

```
        if len(hottest) > 5:
            hottest = sorted(hottest, key=operator.itemgetter(0), reverse=True)
            hottest.pop(len(hottest) - 1)

    elif metric == 'TMIN':
        avg_min += value
        min_count += 1
        if min[0] > value:
            min = (value,id,date)
        coldest.append((value,id,date))

        if len(coldest) > 5:
            coldest = sorted(coldest, key=operator.itemgetter(0))
            coldest.pop(len(coldest) - 1)

print 'Year: %s' % current_year
print 'Average TMAX: %s' % (avg_max * 1.0 / max_count)
print 'Average TMIN: %s' % (avg_min * 1.0 / min_count)
print 'Max TMAX: val: %s' % (hottest[0][0])
print 'Min TMIN: val: %s' %(coldest[0][0])
print 'Hottest Stations %s' % ([x[1] for x in hottest])
print 'Hottest Station Values %s' % ([x[0] for x in hottest])
print 'Coldest Stations %s' % ([x[1] for x in coldest])
print 'Coldest Station Values %s' % ([x[0] for x in coldest])
print
'================================================================================'
print 'Hottest Day: %s | value: %s | Station: %s' % (max[2],max[0],max[1])
```

```
print 'Coldest Day: %s | value: %s | Station: %s' % (min[2],min[0],min[1])
```