Analyzing Large Climate Datasets using MapReduce
Programming Assignment #5
CS677 High Performance Computing
December 16, 2020
Andrew Weston

Abstract: It has been speculated that, due to climate change, global weather is entering a period of increased volatility. We present a Map-Reduce solution to analyzing a National Climatic Data Center weather data set achieving an 8.3x Speedup for some program implementations. We also analyze yearly statistics for an “early” and a “late” sample - the years 1980-1989 and 2003-2012 respectively. We find some narrow, limited evidence supporting the hypothesis of an increase in weather volatility with respect to air temperature in particular.

1. Introduction

1.1 Data-Intensive Computing

The amount of data constantly being collected by automated sensors is staggeringly large, and growing; for example, weather data acquired for a single year amounts to terabytes of information. It is not uncommon for the amount of data required for a particular analysis or experiment to be larger than the capacity of a single storage device to hold it. This is just one of the reasons that Big Data is increasingly stored “in the cloud”. To analyze that data, the distributed nature of cloud computing has engendered a form of parallel processing known as the MapReduce paradigm. The use of the Hadoop/Spark platform in support of the MapReduce paradigm has allowed for sophisticated and potentially time-consuming analyses to be conducted in parallel, in the cloud.

1.2 National Climatic Data Center Weather Observations

The National Oceanic and Atmospheric Administration (NOAA) runs an organization called the National Climatic Data Center (NCDC) which maintains a set of weather observations obtained from automated weather stations located in the United States. This weather observations data is made available to the public and we consider the dataset from 1980-2012.

Each year is composed of over 10,000 files, stored as 1-5 GB of compressed data. Each file provides key weather data readings (e.g. air temperature, wind speed, barometric pressure, etc.) reported from sensor stations across the United States. Each line in a file contains a single time-stamped sensor reading. The data is carefully formatted (see the Data Format in Appendix A). Appendix A describes the exact position within each line of the desired information, along with any codes required to interpret it. For example, Figure 1 below is from one line in a file. Examining the values at positions 16-23 (highlighted) reveals that this set of sensor readings was taken on January 1st, 2012 (YYYYMMDD).

Figure 1

```
018801001099999201201010004+70933-008667FM-12+0009ENJAV0201001N006010021019N0030001N1+
00171+0 0121096611ADDAA106002091AY161061AY221061G
F108991081071002501999999MA1999999096501MD1710
221+9999MW1611REMSYN088AAXX0100101001113308100610017200123965049661570226002176162887/
33391109;
```

Notably, sensor readings may occasionally have missing values; this is typically indicated by a special code (e.g. ‘9999’) in the designated positions within the line. Each weather reading is also followed by a “quality code”, describing the reliability of the reading.

1.3 Climate Change and Global Weather

It has been speculated that, due to climate change, global weather is entering a period of increased volatility (e.g. drastic temperature extremes, more forceful storms). Is there evidence of this in the weather readings recorded across the United States in recent years? What other trends can be observed? We can compare weather sensor metrics from the 1980’s with those from the 2000’s as a rudimentary test of this hypothesis. We emphasize that such a comparison is in no way sufficient to capture the complex realities of such a multi-variable phenomenon as climate change. Nonetheless, this experiment illustrates how scientists can begin to use Big Data to validate models and test hypotheses.

2. Solutions using Map-Reduce

First, we summarize the Map-Reduce framework in general. Next, we explain how we applied the Map-Reduce framework to compute one arbitrary summary statistic: minimum air temperature. Finally, we present our straightforward validation strategy.

2.1 Map-Reduce

Map-Reduce is a programming framework that uses automatic parallelization and distribution to perform computation on large data sets in a fault-tolerant manner. Map-Reduce programs “map” the computational workload by designating parallel tasks. Next the Map-Reduce implementation (e.g. Hadoop, Spark, etc.) performs an intermediate “shuffle” to relocate and group data for more efficient reduction. Finally, a Map-Reduce program “reduces” this grouped data to meaningful output based on a specified reducer function.

2.2 Map-Reduce Solution to Minimum Air Temperature Computation

Here we explain how we applied the Map-Reduce framework to compute one particular type of summary statistic, the minimum air temperature, for a large data set (e.g. all of the reported weather observations for the year 1980). We applied a substantially similar approach to other statistics (e.g. max and average) as well as to other data fields (e.g. wind speed and air pressure).

Appendix B shows the source code of the program. Line 288 invokes a helper function, compute_air_temp_stats, with the Resilient Distributed Dataset (RDD) representing the weather observations for a particular year. The helper function creates a new RDD by filtering air temperature observations with "suspect" or "erroneous" quality codes. Additionally, missing values are also filtered.

Once the RDD is cleaned, we invoke the air_temp_min_mapper defined on Line 27. This function takes a line of input as a parameter and returns a key-value pair where the key is "air_temp_min" and the value is the slice of the line of input representing that value. We use constants to represent what values are being extracted from the RDD. Thereafter, we reduce the results by their keys. The reducer function "min_reducer" simply returns the lesser of two parameters. Since all of the air_temp_min keys are the same this produces an RDD-wide, global minimum result.

2.3 Solution Validation

Our sequential program running on a single host and our distributed program are exactly the same. The difference is that the distributed computation is parallelized across the virtual cluster by the Map-Reduce implementation. We simply ran a "diff" command to verify that the result of both trials produced the same results (save differences in runtime). Appendix C shows sample output for the year 1980. Full details of the results for 1980-2012 are presented in Appendix D.

3. Experimental Results

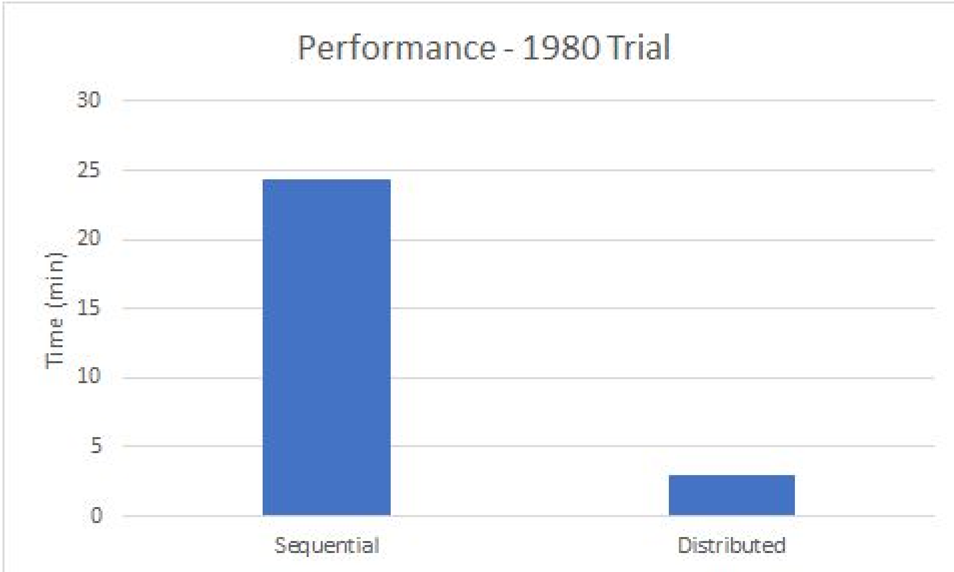
We determined experimental results on a Grand Valley State University Architecture Laboratory virtual cloud. The system included 10 machines - each with an AMD Ryzen 7 2700x 8-core CPU (16 logical cores) with 16 GB RAM. Additionally, every machine ran Ubuntu 20.04.

Additionally, we note that Speedup is defined by the equation:

$$Speedup = Time_{Sequential} / Time_{Parallel}$$

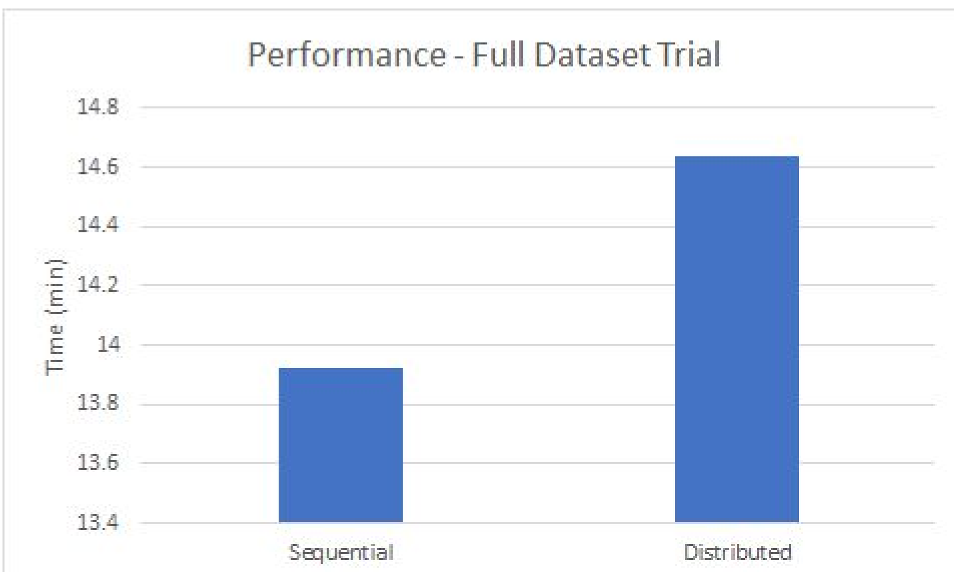
We executed two trials for both solutions: one trial printed the results of the computation and the other "time to solution" version did not. The program that printed achieved an 8.3x Speedup compared to the sequential version. In contrast, the time to solution trials' runtimes were close. The distributed program actually slowed down with a "Speedup" factor of 0.95x. Figures 2 and 3 below show the corresponding performance results.

Figure 2



Note: the problem size of the printed output results represents only the year 1980 to make the sequential trial feasible.

Figure 3

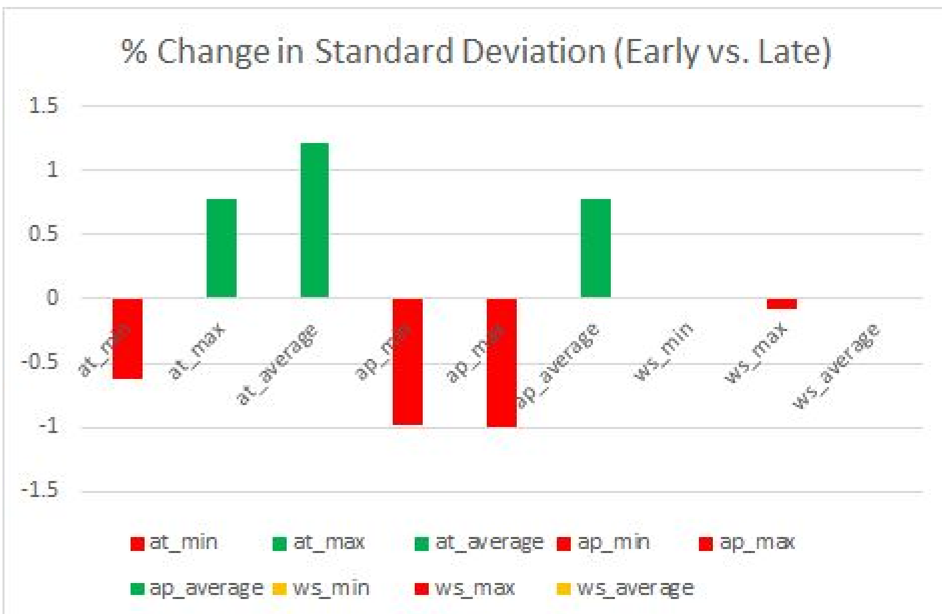


4. Findings

We re-emphasize that our project is a rudimentary climate analysis and extricating the impact of such a complex issue is remarkably challenging. Given the intentional limitations of our experiment, we find some evidence suggesting increased volatility (e.g. drastic temperature extremes, more forceful storms).

After we used Map-Reduce to reduce the total dataset to the annual minimum, maximum, and average values for air temperature, air pressure, and wind speed (shown in Appendix D) we grouped the years 1980-1989 into an “early” sample and the years 2003-2012 into a “late” sample. Then we calculated the squared difference, variance, and standard deviation. Finally, we took the difference between the standard deviations of the late and early sample as a percentage of the early standard deviation to make the more comparable summary in Figure 4.

Figure 4



The green bars support the hypothesis of an increase in weather volatility for maximum air temperature, average air temperature, and average air pressure. Likewise, the red bars do not support the hypothesis with respect to minimum air temperature, and minimum or maximum air pressure. We also note the minimum and average wind speed values are virtually zero.

This limited result seems plausible. The most straightforward causal logic for a high percentage change in standard deviation for any of the three weather observation data fields is that global warming increases the volatility of air temperatures. Why it might impact the maximum temperature but not the minimum temperature could be another research question.

Figures 5 through 7 summarize the standard deviations for each type of weather observation data.

Figure 5

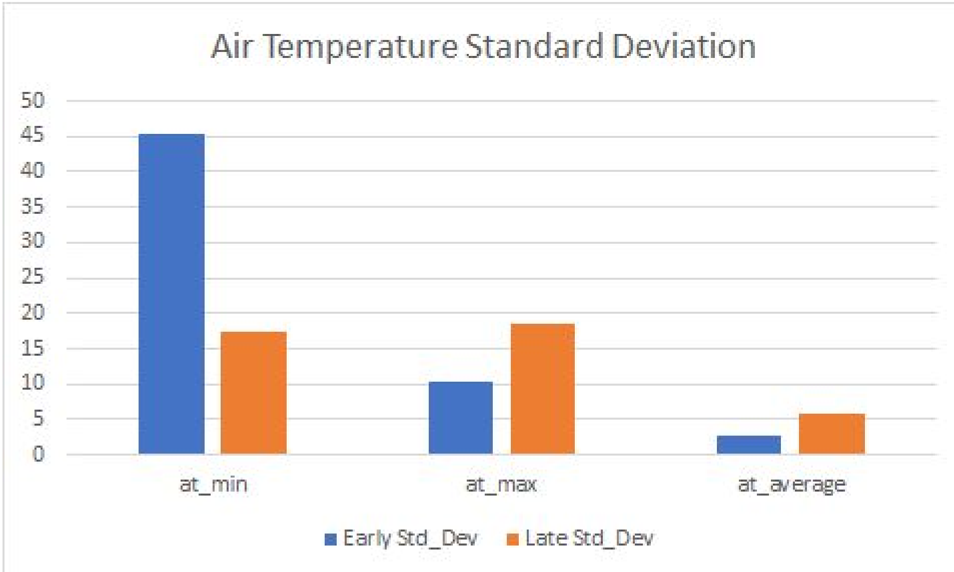


Figure 6

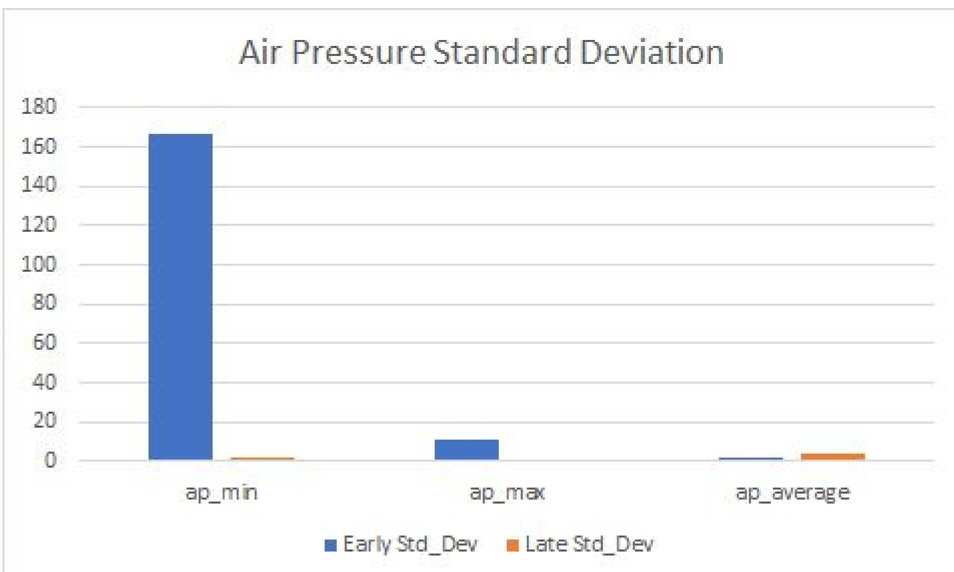
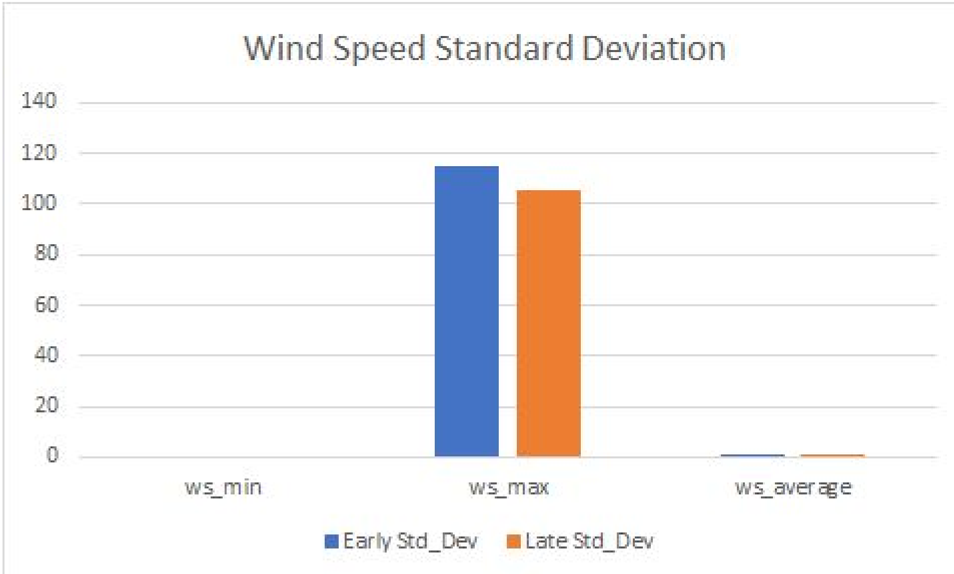


Figure 7



5. Problems and Solutions

5.1 Slowdown - I spent a fair amount of time troubleshooting my distributed slowdown, but am unsure of the source. I believe I had the cloud configured properly. I tried monitoring the Spark dashboard but did not see any clues to the problem. My intuition was that the Map-Reduce program itself was fairly straightforward so that was probably not the source, but I would have liked to have spent more time investigating that. My final theory is there is some issue with how my program used the Arch lab shared file system.

5.2 Enhancement Using Due Date

Appendix B Line 299 shows an extension to the computation explained in Sections 2 and 3. The RDD b_day that filters all weather observations except for those on April 30th represents my wife's due date. I intended to compute the summary statistics using only April 30th observations and then to use a single exponential smoothing function to try to "predict" the statistics for my son's birthday. Unfortunately, I was unable to complete the computation due to time constraints.

Appendix A - NOAA NCDC Data Format

Control Data Section

POS: 1-4

TOTAL-VARIABLE-CHARACTERS (this includes remarks, additional data, and element quality section)
The number of characters in the variable data section. The total record length = 105 + the value stored in this field.
DOM: A general domain comprised of the characters in the ASCII character set.
MIN: 0000 MAX: 9999

POS: 5-10

FIXED-WEATHER-STATION USAF MASTER STATION CATALOG identifier
The identifier that represents a FIXED-WEATHER-STATION.
DOM: A general domain comprised of the characters in the ASCII character set.
COMMENT: This field includes all surface reporting stations, including ships, buoys, etc.

POS: 11-15

FIXED-WEATHER-STATION NCDC WBAN identifier
The identifier that represents a FIXED-WEATHER-STATION.
MIN: 00000 MAX: 99999
DOM: A general domain comprised of the numeric characters (0-9).
COMMENT: This field includes all surface reporting stations, including ships, buoys, etc.

NOTE:
1) For data files obtained via FTP or from NCDC's archive, the filename convention uses the USAF identifier and the WBAN identifier in the filename—eg, 723150-03812-year (such as 2006).
2) As additional data sources are integrated into ISD, the 2 station number fields will be used as an 11-digit ID field, with the first 2 digits representing the WMO block number (if applicable).

POS: 16-23

GEOPHYSICAL-POINT-OBSERVATION date
The date of a GEOPHYSICAL-POINT-OBSERVATION.
MIN: 00000101 MAX: 99991231
DOM: A general domain comprised of integer values 0-9 in the format YYYYMMDD.
YYYY can be any positive integer value; MM is restricted to values 01-12; and DD is restricted to values 01-31.

POS: 24-27

GEOPHYSICAL-POINT-OBSERVATION time
The time of a GEOPHYSICAL-POINT-OBSERVATION based on Coordinated Universal Time Code (UTC).
MIN: 0000 MAX: 2359
DOM: A general domain comprised of integer values 0-9 in the format HHMM.
HH is restricted to values 00-23; MM is restricted to values 00-59.

POS: 28-28

GEOPHYSICAL-POINT-OBSERVATION data source flag
The flag of a GEOPHYSICAL-POINT-OBSERVATION showing the source or combination of sources used in creating the observation.
MIN: 1 MAX: Z
DOM: A general domain comprised of values 1-9 and A-E.
1 = USAF SURFACE HOURLY observation, candidate for merge with NCDC SURFACE HOURLY (not yet merged, failed element cross-checks)
2 = NCDC SURFACE HOURLY observation, candidate for merge with USAF SURFACE HOURLY (not yet merged, failed element cross-checks)
3 = USAF SURFACE HOURLY/NCDC SURFACE HOURLY merged observation
4 = USAF SURFACE HOURLY observation
5 = NCDC SURFACE HOURLY observation
6 = ASOS/AWOS observation from NCDC
7 = ASOS/AWOS observation merged with USAF SURFACE HOURLY observation
8 = MAPSO observation (NCDC)
A = USAF SURFACE HOURLY/NCDC HOURLY PRECIPITATION merged observation, candidate for merge with NCDC SURFACE HOURLY (not yet merged, failed element cross-checks)
B = NCDC SURFACE HOURLY/NCDC HOURLY PRECIPITATION merged observation, candidate for merge with USAF SURFACE HOURLY (not yet merged, failed element cross-checks)
C = USAF SURFACE HOURLY/NCDC SURFACE HOURLY/NCDC HOURLY PRECIPITATION merged observation

D = USAF SURFACE HOURLY/NCDC HOURLY PRECIPITATION merged observation
E = NCDC SURFACE HOURLY/NCDC HOURLY PRECIPITATION merged observation
F = Form OMR/1001 – Weather Bureau city office (keyed data)
G = SAO surface airways observation, pre-1949 (keyed data)
H = SAO surface airways observation, 1965-1981 format/period (keyed data)
I = Climate Reference Network observation
J = Cooperative Network observation
K = Radiation Network observation
L = Data from Climate Data Modernization Program (CDMP) data source
N = NCAR / NCDC cooperative effort (various national datasets)
9 = Missing

Note: Latitude, longitude, elevation, and call letters for some locations with data from multiple sources (see data source flag above) will sometimes vary within a data file due to differences in the metadata from the originating source. This does not indicate that the station locations differ; only that the metadata have not yet been fully reflected in the data records.

POS: 29-34
GEOPHYSICAL-POINT-OBSERVATION latitude coordinate
The latitude coordinate of a GEOPHYSICAL-POINT-OBSERVATION where southern hemisphere is negative.
MIN: -90000 MAX: +90000
UNITS: Angular Degrees
SCALING FACTOR: 1000
DOM: A general domain comprised of the numeric characters (0-9), a plus sign (+), and a minus sign (-).
+99999 = Missing

POS: 35-41
GEOPHYSICAL-POINT-OBSERVATION longitude coordinate
The longitude coordinate of a GEOPHYSICAL-POINT-OBSERVATION where values west from 000000 to 179999 are signed negative.
MIN: -179999 MAX: +180000 UNITS: Angular Degrees
SCALING FACTOR: 1000
DOM: A general domain comprised of the numeric characters (0-9), a plus sign (+), and a minus sign (-).
+999999 = Missing

POS: 42-46
GEOPHYSICAL-REPORT-TYPE code
The code that denotes the type of geophysical surface observation.
DOM: A specific domain comprised of the characters in the ASCII character set.
FM-12 = SYNOP Report of surface observation form a fixed land station
FM-13 = SHIP Report of surface observation from a sea station
FM-14 = SYNOP MOBIL Report of surface observation from a mobile land station
FM-15 = METAR Aviation routine weather report
FM-16 = SPECI Aviation selected special weather report
FM-18 = BUOY Report of a buoy observation
SAO = Airways report (includes record specials)
SAOSP = Airways special report (excluding record specials)
AERO = Aerological report
AUTO = Report from an automatic station
SY-AE = Synoptic and aero merged report
SY-SA = Synoptic and airways merged report
SY-MT = Synoptic and METAR merged report
SY-AU = Synoptic and auto merged report
SA-AU = Airways and auto merged report
S-S-A = Synoptic, airways, and auto merged report
BOGUS = Bogus report
SMARS = Supplementary airways station report
SOD = Summary of day report from U.S. ASOS or AWOS station
SOM = Summary of month report from U.S. ASOS or AWOS station
WBO = Weather Bureau Office
COOPD = US Cooperative Network summary of day report
COOPS = US Cooperative Network soil temperature report
PCP15 = US 15-minute precipitation network report
PCP60 = US 60-minute precipitation network report
CRN05 = Climate Reference Network report, with 5-minute reporting interval
CRN15 = Climate Reference Network report, with 15-minute reporting interval
SURF = Surface Radiation Network report
BRAZ = Dataset from Brazil

GREEN = Dataset from Greenland
AUST = Dataset from Australia
MEXIC = Dataset from Mexico
CRB = Climate Reference Book data from CDMP
WNO = Washington Naval Observatory
NSRDB = National Solar Radiation Data Base
99999 = Missing

POS: 47-51

GEOPHYSICAL-POINT-OBSERVATION elevation dimension
The elevation of a GEOPHYSICAL-POINT-OBSERVATION relative to Mean Sea Level (MSL).
MIN: -0400 MAX: +8850 UNITS: Meters
SCALING FACTOR: 1
DOM: A general domain comprised of the numeric characters (0-9), a minus sign (-), and a plus sign (+).
+9999 = Missing

POS: 52-56

FIXED-WEATHER-STATION call letter identifier
The identifier that represents the call letters assigned to a FIXED-WEATHER-STATION.
DOM: A general domain comprised of the characters in the ASCII character set.
99999 = Missing.

POS: 57-60

METEOROLOGICAL-POINT-OBSERVATION quality control process name
The name of the quality control process applied to a weather observation.
DOM: A general domain comprised of the ASCII character set.

Mandatory Data Section

Bold type below indicates that the element may include data originating from NCDC's NCDC SURFACE HOURLY/ASOS/AWOS or from AFCCC's USAF SURFACE HOURLY. Otherwise, data originated from USAF SURFACE HOURLY.

Note: For the quality code fields with each data element, the following may appear in data which were processed through NCDC's Interactive QC system (manual interaction), for selected parameters:
A – Data value flagged as suspect, but accepted as good value.
U – Data value replaced with edited value.
P – Data value not originally flagged as suspect, but replaced by validator.
I – Data value not originally in data, but inserted by validator.
M - Manual change made to value based on information provided by NWS or FAA.
C - Temperature and dew point received from Automated Weather Observing Systems (AWOS) are reported in whole degrees Celsius. Automated QC flags these values, but they are accepted as valid.
3) For the quality code fields with each data element, the following may appear in data where the recomputed value replaced the original data value, using an automated process:
R - Data value replaced with value computed by NCDC software.

POS: 61-63

WIND-OBSERVATION direction angle
The angle, measured in a clockwise direction, between true north and the direction from which the wind is blowing.
MIN: 001 MAX: 360 UNITS: Angular Degrees
SCALING FACTOR: 1
DOM: A general domain comprised of the numeric characters (0-9).
999 = Missing. If type code (below) = V, then 999 indicates variable wind direction.

POS: 64-64

WIND-OBSERVATION direction quality code
The code that denotes a quality status of a reported WIND-OBSERVATION direction angle.
DOM: A specific domain comprised of the characters in the ASCII character set.
0 = Passed gross limits check
1 = Passed all quality control checks
2 = Suspect
3 = Erroneous
4 = Passed gross limits check , data originate from an NCDC data source
5 = Passed all quality control checks, data originate from an NCDC data source
6 = Suspect, data originate from an NCDC data source
7 = Erroneous, data originate from an NCDC data source
9 = Passed gross limits check if element is present

POS: 65-65

WIND-OBSERVATION type code
The code that denotes the character of the WIND-OBSERVATION.
DOM: A specific domain comprised of the characters in the ASCII character set.
A: Abridged Beaufort
B: Beaufort
C: Calm
H: 5-Minute Average Speed
N: Normal
R: 60-Minute Average Speed
Q: Squall
T: 180 Minute Average Speed
V: Variable
9 = Missing
NOTE: If a value of 9 appears with a wind speed of 0000, this indicates calm winds.

POS: 66-69

WIND-OBSERVATION speed rate

The rate of horizontal travel of air past a fixed point.

MIN: 0000 MAX: 0900 UNITS: meters per second

SCALING FACTOR: 10

DOM: A general domain comprised of the numeric characters (0-9).

9999 = Missing.

POS: 70-70

WIND-OBSERVATION speed quality code

The code that denotes a quality status of a reported WIND-OBSERVATION speed rate.

DOM: A specific domain comprised of the characters in the ASCII character set.

0 = Passed gross limits check
1 = Passed all quality control checks
2 = Suspect
3 = Erroneous
4 = Passed gross limits check , data originate from an NCDC data source
5 = Passed all quality control checks, data originate from an NCDC data source
6 = Suspect, data originate from an NCDC data source
7 = Erroneous, data originate from an NCDC data source
9 = Passed gross limits check if element is present

POS: 71-75

SKY-CONDITION-OBSERVATION ceiling height dimension

The height above ground level (AGL) of the lowest cloud or obscuring phenomena layer aloft with 5/8 or more summation total sky cover, which may be predominantly opaque, or the vertical visibility into a surface-based obstruction. Unlimited = 22000.

MIN: 00000 MAX: 22000 UNITS: Meters

SCALING FACTOR: 1

DOM: A general domain comprised of the numeric characters (0-9).
99999 = Missing.

POS: 76-76

SKY-CONDTION-OBSERVATION ceiling quality code

The code that denotes a quality status of a reported ceiling height dimension.

DOM: A specific domain comprised of the characters in the ASCII character set.
0 = Passed gross limits check
1 = Passed all quality control checks
2 = Suspect
3 = Erroneous
4 = Passed gross limits check , data originate from an NCDC data source
5 = Passed all quality control checks, data originate from an NCDC data source
6 = Suspect, data originate from an NCDC data source
7 = Erroneous, data originate from an NCDC data source
9 = Passed gross limits check if element is present

POS: 77-77

SKY-CONDITION-OBSERVATION ceiling determination code

The code that denotes the method used to determine the ceiling.

DOM: A specific domain comprised of the characters in the ASCII character set.
A: Aircraft
B: Balloon
C: Statistically derived
D: Persistent cirriform ceiling (pre-1950 data)
E: Estimated
M: Measured
P: Precipitation ceiling (pre-1950 data)
R: Radar
S: ASOS augmented
U: Unknown ceiling (pre-1950 data)
V: Variable ceiling (pre-1950 data)
W: Obscured
9: Missing

POS: 78-78

SKY-CONDITION-OBSERVATION CAVOK code

The code that represents whether the 'Ceiling And Visibility Okay' (CAVOK) condition has been reported.

DOM: A specific domain comprised of the characters in the ASCII character set.
N: No
Y: Yes

POS: 79-84

VISIBILITY-OBSERVATION distance dimension

The horizontal distance at which an object can be seen and identified.

MIN: 000000 MAX: 160000 UNITS: Meters

DOM: A general domain comprised of the numeric characters (0-9).

Missing = 999999

NOTE: Values greater than 160000 are entered as 160000

POS: 85-85

VISIBILITY-OBSERVATION distance quality code

The code that denotes a quality status of a reported distance of a visibility observation.

DOM: A specific domain comprised of the characters in the ASCII character set.

0 = Passed gross limits check
1 = Passed all quality control checks
2 = Suspect
3 = Erroneous
4 = Passed gross limits check , data originate from an NCDC data source
5 = Passed all quality control checks, data originate from an NCDC data source
6 = Suspect, data originate from an NCDC data source
7 = Erroneous, data originate from an NCDC data source
9 = Passed gross limits check if element is present

POS: 86-86

VISIBILITY-OBSERVATION variability code

The code that denotes whether or not the reported visibility is variable.

DOM: A specific domain comprised of the characters in the ASCII character set.

N: Not variable
V: Variable
9 = Missing

POS: 87-87

VISIBILITY-OBSERVATION quality variability code

The code that denotes a quality status of a reported VISIBILITY-OBSERVATION variability code.

DOM: A specific domain comprised of the characters in the ASCII character set.

0 = Passed gross limits check
1 = Passed all quality control checks
2 = Suspect
3 = Erroneous
4 = Passed gross limits check , data originate from an NCDC data source
5 = Passed all quality control checks, data originate from an NCDC data source
6 = Suspect, data originate from an NCDC data source
7 = Erroneous, data originate from an NCDC data source
9 = Passed gross limits check if element is present

POS: 88-92

AIR-TEMPERATURE-OBSERVATION air temperature

The temperature of the air.

MIN: -0932 MAX: +0618 UNITS: Degrees Celsius

SCALING FACTOR: 10

DOM: A general domain comprised of the numeric characters (0-9), a plus sign (+), and a minus sign (-).

+9999 = Missing.

POS: 93-93

AIR-TEMPERATURE-OBSERVATION air temperature quality code

The code that denotes a quality status of an AIR-TEMPERATURE-OBSERVATION.

DOM: A specific domain comprised of the characters in the ASCII character set.

0 = Passed gross limits check
1 = Passed all quality control checks
2 = Suspect
3 = Erroneous
4 = Passed gross limits check , data originate from an NCDC data source
5 = Passed all quality control checks, data originate from an NCDC data source
6 = Suspect, data originate from an NCDC data source
7 = Erroneous, data originate from an NCDC data source
9 = Passed gross limits check if element is present
A = Data value flagged as suspect, but accepted as a good value
C = Temperature and dew point received from Automated Weather Observing System (AWOS) are reported in whole degrees Celsius. Automated QC flags these values, but they are accepted as valid.
I = Data value not originally in data, but inserted by validator
M = Manual changes made to value based on information provided by NWS or FAA
P = Data value not originally flagged as suspect, but replaced by validator
R = Data value replaced with value computed by NCDC software
U = Data value replaced with edited value

POS: 94-98

AIR-TEMPERATURE-OBSERVATION dew point temperature

The temperature to which a given parcel of air must be cooled at constant pressure and water vapor content in order for saturation to occur.

MIN: -0982 MAX: +0368 UNITS: Degrees Celsius

SCALING FACTOR: 10

DOM: A general domain comprised of the numeric characters (0-9), a plus sign (+), and a minus sign (-).

+9999 = Missing.

POS: 99-99

AIR-TEMPERATURE-OBSERVATION dew point quality code

The code that denotes a quality status of the reported dew point temperature.

DOM: A specific domain comprised of the characters in the ASCII character set.

0 = Passed gross limits check
1 = Passed all quality control checks
2 = Suspect
3 = Erroneous
4 = Passed gross limits check , data originate from an NCDC data source
5 = Passed all quality control checks, data originate from an NCDC data source
6 = Suspect, data originate from an NCDC data source
7 = Erroneous, data originate from an NCDC data source
9 = Passed gross limits check if element is present
A = Data value flagged as suspect, but accepted as a good value
C = Temperature and dew point received from Automated Weather Observing System (AWOS) are reported in whole degrees Celsius. Automated QC flags these values, but they are accepted as valid.
I = Data value not originally in data, but inserted by validator
M = Manual changes made to value based on information provided by NWS or FAA
P = Data value not originally flagged as suspect, but replaced by validator
R = Data value replaced with value computed by NCDC software
U = Data value replaced with edited value

POS: 100-104

ATMOSPHERIC-PRESSURE-OBSERVATION sea level pressure

The air pressure relative to Mean Sea Level (MSL).

MIN: 08600 MAX: 10900 UNITS: Hectopascals

SCALING FACTOR: 10

DOM: A general domain comprised of the numeric characters (0-9).

99999 = Missing.

POS: 105-105

ATMOSPHERIC-PRESSURE-OBSERVATION sea level pressure quality code

The code that denotes a quality status of the sea level pressure of an ATMOSPHERIC-PRESSURE-OBSERVATION.

DOM: A specific domain comprised of the characters in the ASCII character set.

0 = Passed gross limits check
1 = Passed all quality control checks
2 = Suspect
3 = Erroneous
4 = Passed gross limits check , data originate from an NCDC data source
5 = Passed all quality control checks, data originate from an NCDC data source
6 = Suspect, data originate from an NCDC data source
7 = Erroneous, data originate from an NCDC data source
9 = Passed gross limits check if element is present

Appendix B - NOAA NCDC Map-Reduce Source Code

```python
from __future__ import print_function
from pyspark import SparkContext
import timeit

AIR_TEMP_START = 87
AIR_TEMP_END = 92
AIR_TEMP_QUALITY_CODE = 92

AIR_PRESSURE_START = 99
AIR_PRESSURE_END = 104
AIR_PRESSURE_QUALITY_CODE = 104

WIND_SPEED_START = 65
WIND_SPEED_END = 69
WIND_SPEED_QUALITY_CODE = 69

"""
 This mapper function maps the parameter input line of NOAA climate
 data to the key-value pair of 'air_temp_min' and the corresponding
 minimum air temperature value encoded in the input line.

 @param line is the NOAA climate data sample.
 @return key-value pair of 'air_temp_min' to temperature
"""


def air_temp_min_mapper(line):
    return ('air_temp_min', int(line[AIR_TEMP_START:AIR_TEMP_END]))


"""
 This mapper function maps the parameter input line of NOAA climate
 data to the key-value pair of 'air_temp_max' and the corresponding
 maximum air temperature value encoded in the input line.

 @param line is the NOAA climate data sample.
 @return key-value pair of 'air_temp_max' to temperature
"""


def air_temp_max_mapper(line):
    return ('air_temp_max', int(line[AIR_TEMP_START:AIR_TEMP_END]))


"""
 This mapper function maps the parameter input line of NOAA
```

```
climate
 data to the key-value pair of 'air_temp_avg' and the corresponding
 tuple of the air temperature value encoded in the input line and
 unity representing a single sample.

 @param line is the NOAA climate data sample.
 @return key-value pair of 'air_temp_avg' to tuple temperature, 1
"""


def air_temp_avg_mapper(line):
    return 'air_temp_avg', (int(line[AIR_TEMP_START:AIR_TEMP_END]), 1)


"""
 This mapper function maps the parameter input line of NOAA climate
 data to the key-value pair of 'air_pressure_min' and the corresponding
 minimum air pressure value encoded in the input line.

 @param line is the NOAA climate data sample.
 @return key-value pair of 'air_pressure_min' to pressure
"""


def air_pressure_min_mapper(line):
    return ('air_pressure_min',
            int(line[AIR_PRESSURE_START:AIR_PRESSURE_END]))


"""
 This mapper function maps the parameter input line of NOAA climate
 data to the key-value pair of 'air_pressure_max' and the corresponding
 maximum air pressure value encoded in the input line.

 @param line is the NOAA climate data sample.
 @return key-value pair of 'air_pressure_max' to pressure
"""


def air_pressure_max_mapper(line):
    return ('air_pressure_max',
            int(line[AIR_PRESSURE_START:AIR_PRESSURE_END]))


```

```
"""
 This mapper function maps the parameter input line of NOAA climate
 data to the key-value pair of 'air_pressure_avg' and the corresponding
 tuple of the air pressure value encoded in the input line and
 unity representing a single sample.

 @param line is the NOAA climate data sample.
 @return key-value pair of 'air_pressure_avg' to tuple temperature, 1
"""


def air_pressure_avg_mapper(line):
    return 'air_pressure_avg', \
           (int(line[AIR_PRESSURE_START:AIR_PRESSURE_END]), 1)


"""
 This mapper function maps the parameter input line of NOAA climate
 data to the key-value pair of 'wind_speed_min' and the corresponding
 minimum wind speed value encoded in the input line.

 @param line is the NOAA climate data sample.
 @return key-value pair of 'wind_speed_min' to wind speed
"""


def wind_speed_min_mapper(line):
    return ('wind_speed_min',
            int(line[WIND_SPEED_START:WIND_SPEED_END]))


"""
 This mapper function maps the parameter input line of NOAA climate
 data to the key-value pair of 'wind_speed_max' and the corresponding
 maximum wind speed value encoded in the input line.

 @param line is the NOAA climate data sample.
 @return key-value pair of 'wind_speed_max' to wind speed
"""


def wind_speed_max_mapper(line):
    return ('wind_speed_max',
            int(line[WIND_SPEED_START:WIND_SPEED_END]))

```

```

"""
 This mapper function maps the parameter input line of NOAA climate
 data to the key-value pair of 'wind_speed_avg' and the corresponding
 tuple of the wind speed value encoded in the input line and unity
 representing a single sample.

 @param line is the NOAA climate data sample.
 @return key-value pair of 'wind_speed_avg' to wind speed
"""


def wind_speed_avg_mapper(line):
    return 'wind_speed_avg', \
           (int(line[WIND_SPEED_START:WIND_SPEED_END]), 1)


"""
 This reducer function reduces to the lesser of its two parameter
 values.

 @param x is a comparable value.
 @param y is a comparable value.
 @return is the lesser value.
"""


def min_reducer(x, y):
    return x if (x < y) else y


"""
 This reducer function reduces to the larger of its two parameter
 values.

 @param x is a comparable value.
 @param y is a comparable value.
 @return is the larger value.
"""


def max_reducer(x, y):
    return x if (x > y) else y


"""
 This reducer function reduces two tuples to the sum
```

```
 of their corresponding components.

 @param x is a tuple to be combined.
 @param y is a tuple to be combined.
 @return is the sum tuple.
"""


def avg_reducer(x, y):
    # combine corresponding tuple components
    return (x[0] + y[0], x[1] + y[1])


"""
 This helper function computes aggregate air temperature statistics
 (min, max, and average) for the parameter RDD.  It also filters
 suspect and erroneous samples.

 @param lines is the RDD of weather samples.
"""


def compute_air_temp_stats(lines):
    clean = lines.filter(
        lambda x: x[AIR_TEMP_QUALITY_CODE] != '2' and
                  x[AIR_TEMP_QUALITY_CODE] != '3' and
                  x[AIR_TEMP_QUALITY_CODE] != '6' and
                  x[AIR_TEMP_QUALITY_CODE] != '7' and
                  (x[AIR_TEMP_QUALITY_CODE] != '9' or
                   x[AIR_TEMP_START:AIR_TEMP_END] != '+9999'))
    min_output = clean.map(air_temp_min_mapper) \
        .reduceByKey(min_reducer)
    print(min_output.collect())
    max_output = clean.map(air_temp_max_mapper) \
        .reduceByKey(max_reducer)
    print(max_output.collect())
    avg_output = clean.map(air_temp_avg_mapper) \
        .reduceByKey(avg_reducer)
    print(avg_output.collect())


"""
 This helper function computes aggregate wind speed statistics
 (min, max, and average) for the parameter RDD.  It also filters
 suspect and erroneous samples.

 @param lines is the RDD of weather samples.
"""

```

```

def compute_wind_speed_stats(lines):
    # don't prohibit code 9 for "calm winds" case
    # simply exclude missing values
    clean = lines.filter(lambda x:
                         x[WIND_SPEED_QUALITY_CODE] != '2' and
                         x[WIND_SPEED_QUALITY_CODE] != '3' and
                         x[WIND_SPEED_QUALITY_CODE] != '6' and
                         x[WIND_SPEED_QUALITY_CODE] != '7' and
                         x[WIND_SPEED_START:WIND_SPEED_END] != '9999')
    min_output = clean.map(wind_speed_min_mapper) \
        .reduceByKey(min_reducer)
    print(min_output.collect())
    max_output = clean.map(wind_speed_max_mapper) \
        .reduceByKey(max_reducer)
    print(max_output.collect())
    avg_output = clean.map(wind_speed_avg_mapper) \
        .reduceByKey(avg_reducer)
    print(avg_output.collect())


"""
 This helper function computes aggregate air pressure statistics
 (min, max, and average) for the parameter RDD.  It also filters
 suspect and erroneous samples.

 @param lines is the RDD of weather samples.
"""


def compute_air_pressure_stats(lines):
    clean = lines.filter(
        lambda x: x[AIR_PRESSURE_QUALITY_CODE] != '2' and
                  x[AIR_PRESSURE_QUALITY_CODE] != '3' and
                  x[AIR_PRESSURE_QUALITY_CODE] != '6' and
                  x[AIR_PRESSURE_QUALITY_CODE] != '7' and
                  x[AIR_PRESSURE_START:AIR_PRESSURE_END] != '99999')
    min_output = clean.map(air_pressure_min_mapper) \
        .reduceByKey(min_reducer)
    print(min_output.collect())
    max_output = clean.map(air_pressure_max_mapper) \
        .reduceByKey(max_reducer)
    print(max_output.collect())
    avg_output = clean.map(air_pressure_avg_mapper) \
        .reduceByKey(avg_reducer)
    print(avg_output.collect())


```

```
"""
 This driver function computes aggregate weather statistics (min, max,
 and avg) for various weather data fields (air temperature, air
 pressure, and wind speed) using an RDD of NOAA weather data.  It also
 demonstrates filtering the RDD to an arbitrary date.
"""
if __name__ == "__main__":
    sc = SparkContext(appName="PySparkClimate")
    YEAR_START = 1980
    YEAR_END = 2012
    start_time = timeit.default_timer()
    for directory in range(YEAR_START, YEAR_END + 1):
        lines = sc.textFile('/home/DATA/NOAA_weather/' + str(directory), 1)
        print(directory)
        print("Summary Statistics")
        compute_air_temp_stats(lines)
        compute_air_pressure_stats(lines)
        compute_wind_speed_stats(lines)
        print('B-Day')
        # son's due date is April 30th
        b_day = lines.filter(lambda x: x[19:23] == '0430')
        compute_air_temp_stats(b_day)
        compute_air_pressure_stats(b_day)
        compute_wind_speed_stats(b_day)
    print('Time: ', timeit.default_timer() - start_time)
    sc.stop()

```

Appendix C - Sample Program Output (1980)

```
1980
Summary Statistics
[('air_temp_min', -780)]
[('air_temp_max', 600)]
[('air_temp_avg', (3197100331, 29126545))]
[('air_pressure_min', 8609)]
[('air_pressure_max', 10899)]
[('air_pressure_avg', (199772645495, 19683097))]
[('wind_speed_min', 0)]
[('wind_speed_max', 500)]
[('wind_speed_avg', (1144041554, 30574787))]
B-Day
[('air_temp_min', -740)]
[('air_temp_max', 540)]
[('air_temp_avg', (10152913, 79899))]
[('air_pressure_min', 9448)]
[('air_pressure_max', 10523)]
[('air_pressure_avg', (544629883, 53758))]
[('wind_speed_min', 0)]
[('wind_speed_max', 420)]
[('wind_speed_avg', (3058535, 84027))]
Time:  176.00518232584
```

Appendix D - NOAA NCDC Statistics (1980-2012)

	Year									Birthday								
	air_temp			air_pressure			wind_speed			air_temp			air_pressure			wind_speed		
	min	max	avg	min	max	avg	min	max	avg	min	max	avg	min	max	avg	min	max	avg
1980	-780	600	109.77	8609	10899	10149.45	0	500	37.42	-740	540	127.07	9448	10523	10131.14	0	420	36.39943
1981	-850	580	115.43	9120	10899	10146.61	0	618	37.14	-680	540	131.61	9570	10556	10116.93	0	380	40.88408
1982	-930	617	114.44	8785	10899	10152.4	0	618	36.42	-710	530	122.67	9234	10860	10149.02	0	500	40.01632
1983	-931	616	116.56	8685	10861	10146.51	0	567	37.87	-751	481	133.43	9402	10535	10125.04	0	500	35.96799
1984	-932	617	112.02	8716	10898	10152.04	0	500	37.45	-627	520	123.03	9137	10607	10140.69	0	480	41.05751
1985	-932	611	107.99	8607	10899	10152.42	0	500	37.59	-687	602	127.79	9178	10836	10137.44	0	500	38.45693
1986	-901	607	110.9	8720	10899	10151.07	0	500	37.95	-716	540	133.83	9182	10890	10154.63	0	460	37.86056
1987	-900	607	111.3	8602	10899	10151.24	0	551	37.21	-647	580	141.27	9350	10898	10127.73	0	391	38.73273
1988	-900	607	113.35	9000	10899	10147.29	0	500	38.23	-732	540	124.16	9015	10894	10113.5	0	370	37.6173
1989	-900	606	115.5	8652	10899	10153.76	0	890	37.22	-724	584	123.57	9129	10860	10157.68	0	340	36.03031
1990	-901	607	119.51	8613	10900	10148.48	0	860	38.46	-635	580	132.1	9256	10850	10173.96	0	380	38.94484
1991	-900	607	116.99	8609	10899	10153.59	0	900	37.29	-688	569	124.31	9220	10895	10144.83	0	440	41.90424
1992	-900	605	116.78	8623	10900	10150.09	0	812	37.27	-645	540	128.03	9017	10868	10131.74	0	400	40.38364
1993	-902	567	113.46	8622	10900	10150.17	0	900	37.26	-657	560	136.56	8735	10741	10149.76	0	500	32.66127
1994	-900	568	119.9	8700	10900	10145.53	0	894	37.44	-699	515	133.99	9220	10870	10163.44	0	440	36.71247
1995	-902	567	119.86	8656	10900	10143.88	0	880	37.07	-880	444	134.63	9118	10834	10157.18	0	600	32.33642
1996	-900	561	117.18	8602	10900	10147.75	0	890	36.82	-658	556	130.66	9106	10513	10114.77	0	504	37.73547
1997	-900	565	125.73	8632	10900	10146.39	0	900	35.91	-684	490	139.6	9490	10738	10126.55	0	740	36.58385
1998	-900	568	131.3	8708	10900	10144.31	0	800	36.2	-736	480	147.02	9481	10897	10142.77	0	400	34.3813
1999	-891	568	127.26	8600	10900	10146.77	0	900	35.57	-815	530	135.78	9550	10871	10165.18	0	340	35.13649
2000	-900	568	123.64	8600	10900	10148.22	0	870	35.79	-746	480	143.33	9364	10548	10161.43	0	396	33.56385
2001	-900	568	123.05	8608	10900	10149.62	0	900	37.55	-703	565	146.57	9493	10800	10148.47	0	360	38.05528
2002	-932	568	123.84	8606	10900	10149.35	0	900	36.94	-629	500	130.58	9064	10814	10118.47	0	640	39.7798
2003	-900	565	119.94	8601	10900	10150.34	0	880	35.82	-703	480	139.24	9545	10524	10130.61	0	340	36.58308
2004	-900	567	118.98	8600	10900	10151.34	0	640	36.13	-723	540	137.29	9117	10495	10136.75	0	640	37.48287
2005	-925	610	119.18	8603	10900	10150.32	0	900	35.04	-788	500	124.33	9653	10592	10146.59	0	380	34.59301
2006	-916	610	122.77	8608	10900	10146.54	0	640	35.92	-683	450	127.16	9504	10524	10150.61	0	427	40.14157
2007	-900	610	119.03	8604	10900	10146.61	0	637	35.96	-662	456	146.76	8982	10526	10134.57	0	396	35.56389
2008	-900	610	114.41	8602	10900	10142.6	0	766	36.13	-694	450	120.14	9014	10522	10110.88	0	524	40.14801
2009	-854	610	115.15	8600	10900	10141.76	0	864	35.14	-655	489	137.98	9442	10608	10165.33	0	473	36.17102
2010	-902	617	117.77	8603	10900	10137.62	0	812	34.74	-708	450	136.57	8907	10599	10097.8	0	375	41.08032
2011	-900	618	119.62	8601	10900	10142.56	0	684	35.93	-680	440	129.11	9318	10600	10137.27	0	386	41.79686
2012	-900	600	136.48	8600	10900	10140.5	0	879	35.21	-692	490	145.14	8656	10569	10144.41	0	386	35.30829

Appendix E - % Difference in Standard Deviation Statistics (Early vs. Late)

	Early Std_Dev	Late Std_Dev	Difference	%
at_min	45.41	17.32	-28.09	-0.61859
at_max	10.37	18.43	8.06	0.777242
at_avg	2.65	5.84	3.19	1.203774
ap_min	166.96	2.36	-164.6	-0.98586
ap_max	11.37	0	-11.37	-1
ap_avg	2.51	4.46	1.95	0.776892
ws_min	0	0	0	0
ws_max	114.75	105.06	-9.69	-0.08444
ws_avg	0.48	0.49	0.01	0.020833