

# **A Critical Review of Global Surface Temperature Data Products**

**Ross McKittrick, Ph.D.**  
**Professor of Economics**  
**University of Guelph**  
**Guelph Ontario Canada**

**August 5, 2010**

Version notes:

Original draft July 26 2010.

Revised August 3, 2010 to correct caption and Figure 5-1 - removing dashed horizontal line.

Revised August 4, 2010 to correct caption and Figure 1-10 to compare GHCN raw and adj using only gridcells where both are present.

Revised August 5, 2010 to change Figure 1-7 to show mean absolute latitude; accompanying text also revised.

Comments and corrections are welcome: please send them to [rmckitri@uoguelph.ca](mailto:rmckitri@uoguelph.ca).

## Table of Contents

Table of Contents.....	2
Summary .....	4
1. Land data sources.....	6
1.1. The Global Historical Climatology Network.....	6
1.1.1. Version history.....	6
1.1.2. Description of GHCN v2 archive and updated series.....	6
1.2. GHCN Sample numbers and the drop after 1990 .....	7
1.2.1. Data loss by region.....	10
1.2.2. Growing bias toward airport sources .....	11
1.2.3. Growing bias toward lower latitudes .....	13
1.2.4. Growing bias toward lowland sites.....	13
1.3. Increasing magnitude of adjustments employed to try and fix the problems of sampling discontinuities.....	14
1.4. Summary: GHCN sampling problems. ....	16
2. Dependence of the three major global temperature products on GHCN.....	17
2.1. CRU: more than 98% from GHCN.....	17
2.2. Goddard Institute of Space Studies (GISS): GHCN except Antarctica.....	18
2.3. National Oceanic and Atmospheric Administration (NOAA): GHCN .....	18
2.4. Conclusion re. dependence on GHCN .....	18
3. Input data: Ocean Sources .....	18
3.1. ICOADS: the major input source for all products .....	18
3.2. ICOADS Data Sources.....	19
3.3. Combination of buckets and intake water.....	20
3.4. Sea Ice .....	25
3.5. SST versus air temperature .....	25
3.6. SST data products: Hadley Centre .....	26
3.7. NOAA .....	27
3.8. GISS .....	28
3.9. Conclusions about SST data .....	28
4. A Brief Description of Processing Steps .....	29
4.1. Flowchart .....	30
4.2. Notes on individual methods: GISS, CRU and NOAA .....	31
4.2.1. GISS .....	31
4.2.2. CRU.....	33
4.2.3. NOAA .....	33
4.2.4. Remaining steps .....	33
4.2.5. Conclusions: Processing steps yield minor differences.....	33
4.3. Conclusions about independence of data products .....	33
4.4. Conclusions so far about construction of surface temperature record.....	33
5. Quality of the raw GHCN data .....	34
5.1. Biases related to urbanization and land use change.....	34
5.1.1. General acknowledgment of problem; claims that it has been rectified.....	34
5.1.2. Adjustment methods: CRU .....	36
5.1.3. Adjustment methods: GISS.....	37
5.1.4. Adjustment methods: NOAA.....	39
5.2. <i>Ex post</i> methods for testing the adequacy of the adjustments .....	39
5.2.1. Observations vs reanalysis data .....	39

5.2.2. Calm vs windy conditions ..... 42  
5.2.3. Socioeconomic gridding..... 44  
References..... 46  
Author bio ..... 49  
Appendix: Computational details ..... 50

## Summary

There are three main global temperature histories: the combined CRU-Hadley record (HADCRU), the NASA-GISS (GISTEMP) record, and the NOAA record.

All three global averages depend on the same underlying land data archive, the Global Historical Climatology Network (GHCN). CRU and GISS supplement it with a small amount of additional data. Because of this reliance on GHCN, its quality deficiencies will constrain the quality of all derived products.

The number of weather stations providing data to GHCN plunged in 1990 and again in 2005. The sample size has fallen by over 75% from its peak in the early 1970s, and is now smaller than at any time since 1919. The collapse in sample size has not been spatially uniform. It has increased the relative fraction of data coming from airports to about 50 percent (up from about 30 percent in the 1970s). It has also reduced the average latitude of source data and removed relatively more high-altitude monitoring sites. GHCN applies adjustments to try and correct for sampling discontinuities. These have tended to increase the warming trend over the 20<sup>th</sup> century. After 1990 the magnitude of the adjustments (positive and negative) gets implausibly large.

CRU has stated that about 98 percent of its input data are from GHCN. GISS also relies on GHCN with some additional US data from the USHCN network, and some additional Antarctic data sources. NOAA relies entirely on the GHCN network.

Oceanic data are based on sea surface temperature (SST) rather than marine air temperature (MAT). All three global products rely on SST series derived from the ICOADS archive, though the Hadley Centre switched to a real time network source after 1998, which may have caused a jump in that series. ICOADS observations were primarily obtained from ships that voluntarily monitored SST. Prior to the post-war era, coverage of the southern oceans and polar regions was very thin. Coverage has improved partly due to deployment of buoys, as well as use of satellites to support extrapolation. Ship-based readings changed over the 20<sup>th</sup> century from bucket-and-thermometer to engine-intake methods, leading to a warm bias as the new readings displaced the old. Until recently it was assumed that bucket methods disappeared after 1941, but this is now believed not to be the case, which may necessitate a major revision to the 20<sup>th</sup> century ocean record. Adjustments for equipment changes, trends in ship height, etc., have been large and are subject to continuing uncertainties. Relatively few studies have compared SST and MAT in places where both are available. There is evidence that SST trends overstate nearby MAT trends.

Processing methods to create global averages differ slightly among different groups, but they do not seem to make major differences, given the choice of input data. After 1980 the SST products have not trended upwards as much as land air temperature averages.

The quality of data over land, namely the raw temperature data in GHCN, depends on the validity of adjustments for known problems due to urbanization and land-use change. The adequacy of these adjustments has been tested in three different ways, with two of the three finding evidence that they do not suffice to remove warming biases.

The overall conclusion of this report is that there are serious quality problems in the surface temperature data sets that call into question whether the global temperature history, especially over land, can be considered both *continuous* and *precise*. Users should be aware of these limitations, especially in policy-sensitive applications.



## 1. Land data sources

There are three main global temperature histories: the combined CRU-Hadley record (HADCRU), the NASA-GISS (GISTEMP) record, and the NOAA record.

The purpose of this report is to help users of climate data understand more about how these globally-averaged temperature graphs are created. The main focus is on the construction of the land and ocean data series that go into such graphs. The intent is that readers will understand the uncertainties, sampling problems, discontinuities and other quality problems that are known to many practitioners but which are not typically explained to the public. The overall conclusion of this report is that there are serious quality problems in the surface temperature data sets that call into question whether the global temperature history, especially over land, can be considered sufficiently *continuous* and *accurate* to support the weight of conclusions typically drawn from it.

This section provides an overview of the land-based data used for global temperature archives.

### 1.1. The Global Historical Climatology Network

#### 1.1.1. Version history

All global temperature data products rely heavily on the archive of weather station data known as GHCN – the Global Historical Climatology Network.<sup>1</sup> The GHCN began as a collaboration in the early 1990s between the Carbon Dioxide Information and Analysis Center (CDIAC) and the National Climatic Data Center (NCDC). Its aim was to assemble a more comprehensive temperature data archive than were then available from the CRU or other research units. The first version was released in 1992 (Vose et al. 1992). The data were released on an ‘as-is’ basis with no corrections for inhomogeneities.<sup>2</sup> The second version (GHCN v2) was released in 1997. The construction of GHCN v2 is described in Peterson and Vose (1997) and the quality control methods are described in Peterson et al. (1998). During the preparation of GHCN v2 the authors applied more systematic corrections for inhomogeneities and added metadata such as nearby population and precise information about each station’s location so that users would better understand the source quality better.

#### 1.1.2. Description of GHCN v2 archive and updated series

GHCN v2 compiled 31 different data archives, each with differing amounts of coverage over time. GHCN attempted to obtain original data not subject to homogenization, preferring to give users access to

---

<sup>1</sup> The GHCN Website is <http://www.ncdc.noaa.gov/oa/climate/ghcn-monthly/index.php>. Sources are listed at <http://www.ncdc.noaa.gov/oa/climate/ghcn-monthly/source-table1.html>.

<sup>2</sup> The term ‘inhomogeneity’ when applied to temperature data is rather loosely defined. Its usual meaning is a measurement discontinuity due to a change in equipment, change in the time of observation, relocation of a weather station, etc. Some authors also use it to cover trend biases due to urbanization, land use change and other non-climatic effects, though many authors use different terms for these latter effects. For this reason, when we speak of an archive like GHCN being “corrected for inhomogeneities” this can be interpreted to mean “corrected for measurement discontinuities”, but not necessarily corrected for biases due to local non-climatic effects.

the raw data while developing and applying their own homogeneity corrections. GHCN publishes its raw data as received from sources, as well as a homogeneity-adjusted version (GHCN-adj). Because GHCN compiles data from many different archives, and those archives had, in some cases, each collected the same data, it ended up with large numbers of duplicates. To produce the adjusted version of GHCN the duplicates had to be merged. Sometimes the duplicates have different station ID codes, so numerical search methods had to be used to identify them and combine them into site records. The entire adjustment process involves three stages.

- (i) *Evaluate the source data set quality.* The source had to provide apparently original, raw data and be free of obvious reporting errors.
- (ii) *Examine each series manually.* Each station series was located on Operational Navigation Charts to check the reported elevation, and then the series was compared to expected values based on the known local climatology. Errors discovered at this stage were usually station mislocations, and were corrected if possible, otherwise the record was discarded. Series were also tested for discontinuities by looking for abrupt changes in the mean and standard deviation.
- (iii) *Compare each series to a regional reference series.* For stations with at least 20 years of data, construction of a regional reference series for each station was done as follows.
  - For all stations, compute the first differences in the annual means, in other words the changes from one year to the next in the annual mean.
  - For the selected (“candidate”) station, select the five neighboring stations whose first differences were most highly correlated with those of the candidate station, subject to data availability.
  - Take the average of three of the five neighboring stations to form the reference first difference series.
  - Starting at zero for the first year, form a final reference series by cumulating the reference first difference year by year. An adjustment would also be applied to ensure the final year’s value matches that of the candidate series.

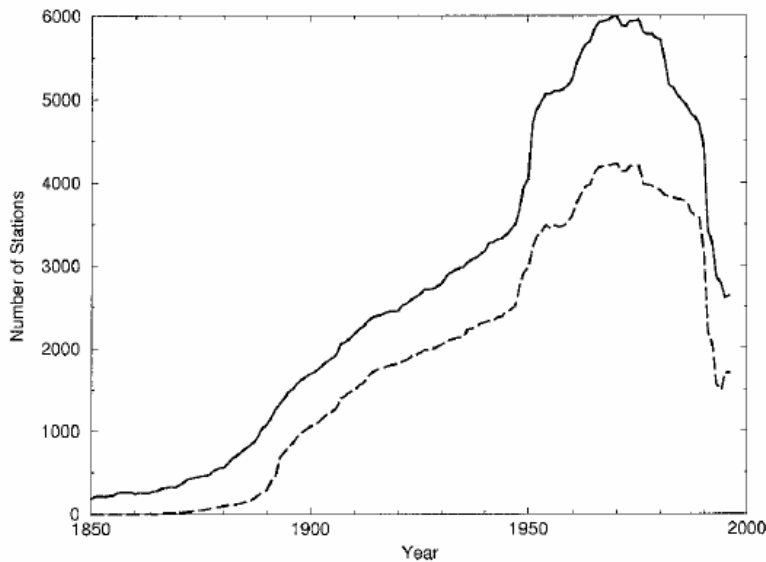
The reference series thus constructed was assumed to represent the climate of the region around the station. The candidate series was then compared to it using regression methods to check for structural breaks (points where the linear trend clearly changes) and other discrepancies.

If a station could not be adjusted using the above method, either because there were too few neighbors or fewer than 20 years of data, then it was not included in the homogeneity-adjusted group, unless the raw data were deemed to be homogeneous. In the case of stations with complete records (including daily mean, maximum and minimum temperature readings), this led to a reduction of the total number of source records from about 8,000 to just under 5,000

## 1.2. GHCN Sample numbers and the drop after 1990

Figure 1-1 below, taken from Peterson and Vose (1997), shows the total number of available weather station records in GHCN. The sample size peaked in the 1960s and 1970s and has fallen dramatically since then.

Climatological records require observations on mean, maximum and minimum temperatures. As shown in Figure 1.1 there are consistently about one-third fewer stations with min-max records compared to mean temperature records.



**Figure 1-1** Number of complete or partial weather station records in GHCN v2. Solid line: mean temperature records. Dashed line: Max/min temperature records. Source: Peterson and Vose (1997) Figure 2a.

While GHCN v2 has at least some data from most places in the world, continuous coverage for the whole of the 20<sup>th</sup> century is largely limited to the US, southern Canada, Europe and a few other locations. Figure 1-2, from Peterson and Vose, contrasts the total availability of stations to the availability of mean temperature records in 1900, and maximum-minimum temperature records back to 1900.

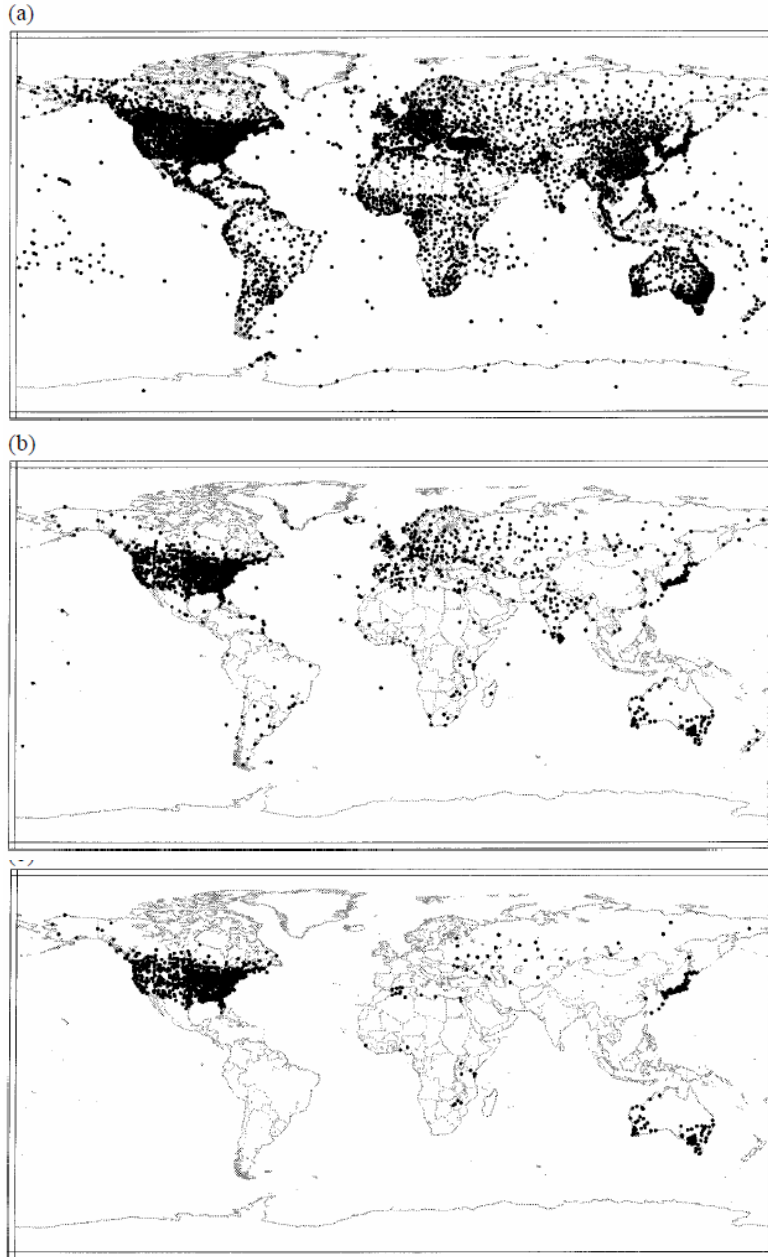
As is clear from the third panel, global coverage is virtually non-existent for maximum-minimum data back to 1900. Apart from the United States, southern Canada and coastal Australia, there are only a few records, with large continental interiors in South America, Africa, Europe and Asia devoid of records.

After the release of GHCN v2 in 1997, only a small minority of stations in the archive were regularly updated. Peterson and Vose (1997) point out that of the 31 source archives used for GHCN, regular monthly updates are available for only three of them. Of these, two are US networks and one is a 1500-station network that automatically reports weather data through the so-called CLIMAT network. The GHCN website does not report which sites are included in the monthly updates, but recent tabulations using GHCN records shows that:

- coverage has fallen everywhere (including the US)
- the sample has become increasingly skewed towards airport sites;
- the sample has migrated from colder latitudes to warmer latitudes;
- the sample has migrated to lower altitudes.

Each of these points will be shown using GHCN data below.





**Figure 1-2**

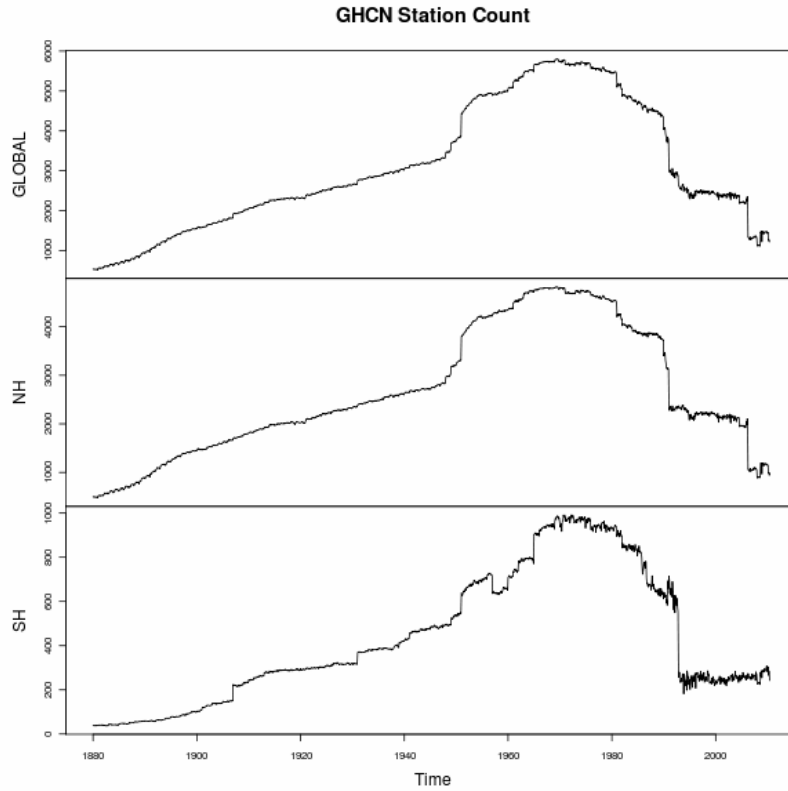
Top panel: locations with at least partial mean temperature records in GHCN v2.

Middle panel: locations with a mean temperature record available in 1900.

Bottom panel: locations with a maximum temperature record available in 1900.

Source: Peterson and Vose (1997), Figures 3 & 4.

As was clear in Figure 1-1 above, the GHCN sample size was falling rapidly by the 1990s. Surprisingly, the decline in GHCN sampling has continued since then. Figure 1-3 shows the total numbers of GHCN weather station records by year. Notice that the drop not only continued after 1989 but became precipitous in 2005. The second and third panels show, respectively, the northern and southern hemispheres, confirming that the station loss has been global. The sample size has fallen by about 75% from its peak in the early 1970s, and is now smaller than at any time since 1919. *As of the present the GHCN samples fewer temperature records than it did at the end of WWI.*

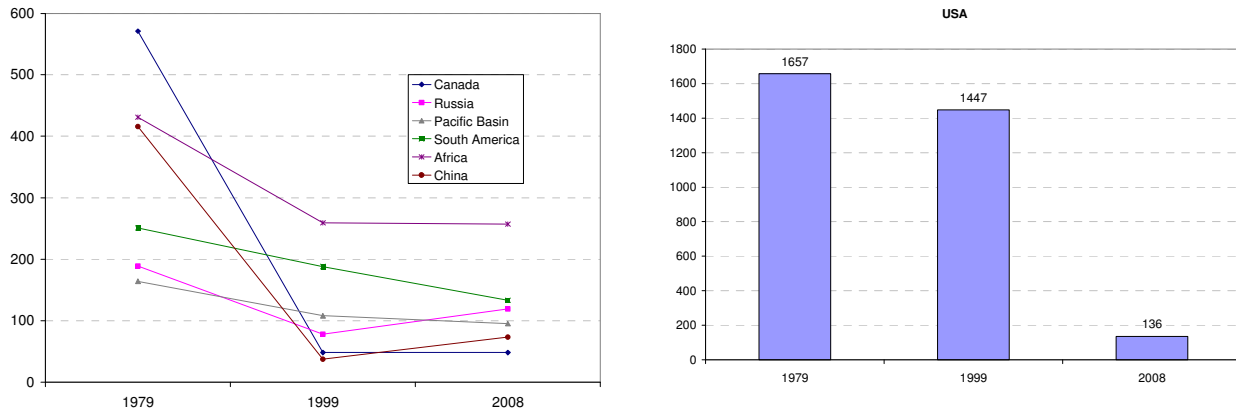


**Figure 1-3:** GHCN Station Count. Top: global; middle: NH; bottom: SH.

Data source: GHCN. See Appendix for calculation details.

### 1.2.1. Data loss by region

The attrition of thermometers has been global. The following chart show the drop in numbers by major geographic region since 1979. The US is shown separately because its dense sampling means it contributed much of the pre-1990 GHCN sample.



**Figure 1-4:** Drop in the numbers of contributing weather stations in the GHCN archive. 1979-2008.

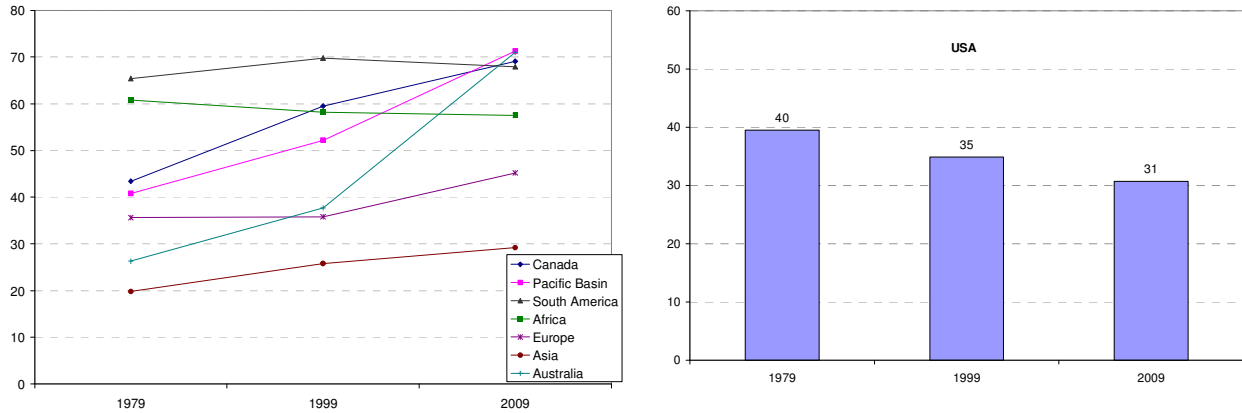
Data source: GHCN; tabulation by <http://chiefio.wordpress.com/2009/11/03/ghcn-the-global-analysis/>.

The drop in GHCN sampling has been alluded to in some publications, but has not been highlighted by sources such as the IPCC. Smith and Reynolds (2005 p. 2023) described the problem as follows (emphasis added):

For LST [Land Surface Temperature], the sampling increase with time is monotonic up to about 1980, **when it begins to decrease slightly**. That recent decrease in LST sampling results from a time lag for the inclusion of some data into the GHCN and station dropouts.

### 1.2.2. Growing bias toward airport sources

The change in the sample was not uniform with respect to source type. For instance it has biased the sample towards airport locations. GHCN had already been heavily-weighted towards airports, which, for many reasons, are not suitable for climatic monitoring. A problem with airports is that they are often in urban or suburban locations that have been built up in the past few decades, and the increase in global air travel has led to increased traffic, pavement, buildings and waste heat, all of which are difficult to remove from the temperature record. As shown in the next chart, the fraction of observations coming from airports increased as a result of the station losses shown above. Most regions were high to begin with: 40 percent or more as of 1980. Now half or more of the measurements from many regions come from airports.



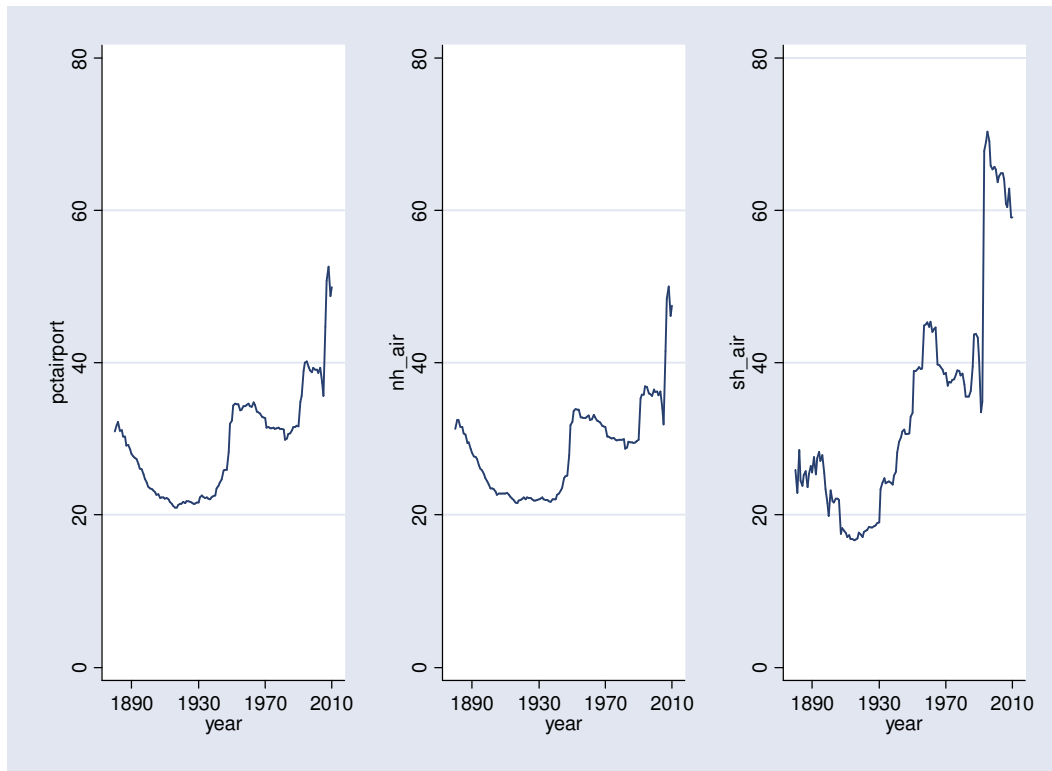
**Figure 1-5:** Change in the percentage of GHCN weather stations located at airports, 1979-2009.

Data source: GHCN; tabulation by <http://chiefio.wordpress.com/2009/12/08/ncdc-gcn-airports-by-year-by-latitude/>.

The US is again shown separately. Just under one-third of the GHCN US data are from airports, which is actually an improvement since 1979, but bear in mind that over the same interval 92% of all US stations were removed from the GHCN sample.

As shown in

Figure 1-6, at the global level, as of 2009 49% of all GHCN data came from airports (46% NH, 59% SH), up from just over 20 percent in the late 1920s.



**Figure 1-6:** Percent GHCN stations located at airports, 1890-2009. Left: global; middle: NH, right: SH.

Data source: GHCN. See Appendix for calculation details.

### 1.2.3. Growing bias toward lower latitudes

The decline in sample has not been spatially uniform. GHCN has progressively lost more and more high latitude sites (e.g. towards the poles) in favour of lower-latitude sites. Other things being equal, this implies less and less data are drawn from remote, cold regions and more from inhabited, warmer regions. As shown in Figure 1-7, mean absolute latitude declined as more stations were added during the 20<sup>th</sup> century. From Figure 1-2 we can see that this came about by the filling in of tropical regions after Europe and the US were already well-sampled. Around 1980 there was a sudden rise, followed by a sudden drop and a period of volatility, and then a second, sharp drop in the mean latitude. This indicates that the drops in sampling were not spatially uniform, that the sample has been volatile, and in recent years it has been biased towards low latitudes.

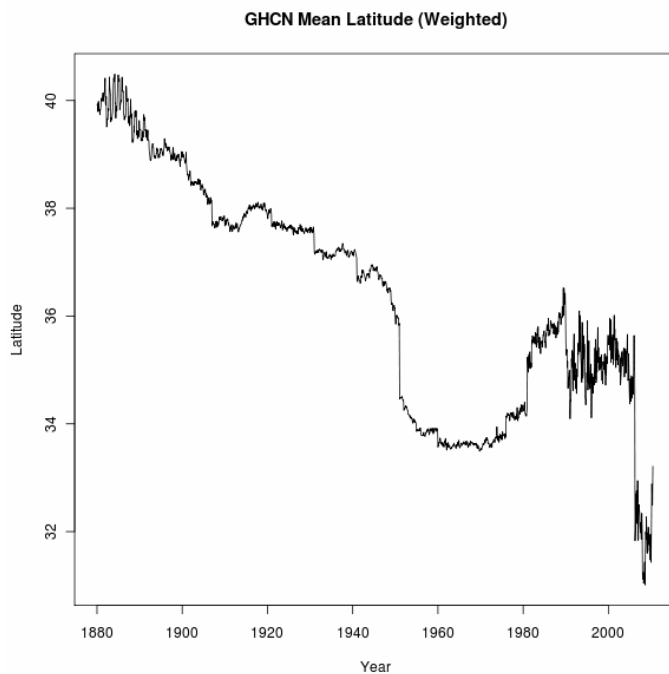


Figure 1-7: GHCN mean absolute latitude of monitoring stations. Data are grouped by latitude band and the bands are weighted by geographical area. Data source: GHCN. See Appendix for calculation details.

### 1.2.4. Growing bias toward lowland sites

Figure 1-8 shows the mean altitude above sea level in the GHCN record. The steady increase is consistent with a move inland of the network coverage, and also increased sampling in mountainous locations. The sample collapse in 1990 is clearly visible as a drop not only in numbers but also in altitude, implying the remote high-altitude sites tended to be lost in favour of sites in valley and coastal locations. This happened a second time in 2005. Since low-altitude sites tend to be more influenced by agriculture, urbanization and other land surface modification, the failure to maintain consistent altitude of the sample detracts from its statistical continuity.

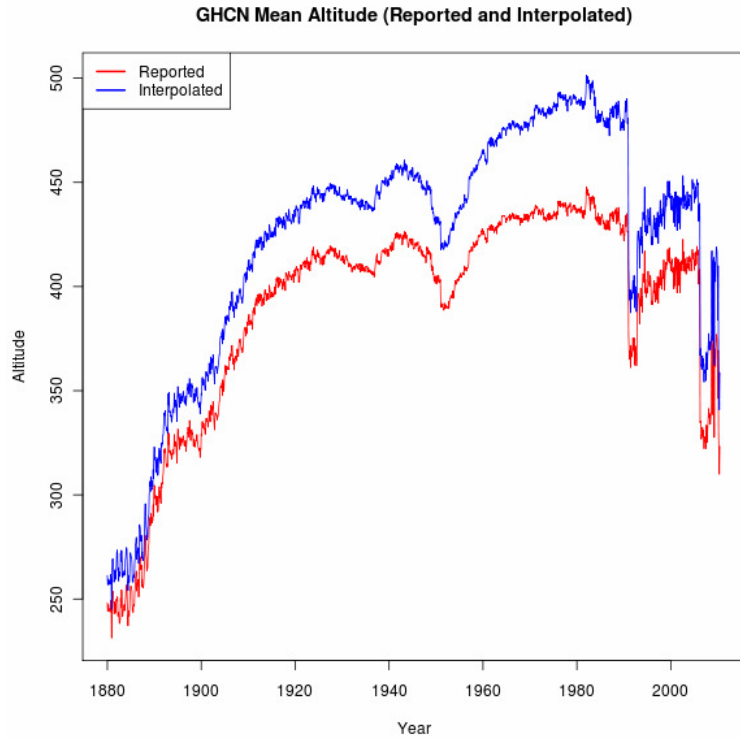


Figure 1-8: Mean altitude above sea level in GHCN records. Red line: raw data. Blue line: including interpolated records. Data source: GHCN. See Appendix for calculation details.

### 1.3. Increasing magnitude of adjustments employed to try and fix the problems of sampling discontinuities

As explained above, GHCN releases two versions of its data: raw and adjusted. The adjustments are intended, among other things to remove the effects of the sorts of discontinuities described above, as well as those arising from changes in equipment and changes in the position of monitoring stations (Peterson et al. 1998). Urban heat island effects are dealt with separately (see Section 5). The following graph compares the unadjusted and adjusted global averages using a standard gridding methodology (see Appendix for details).

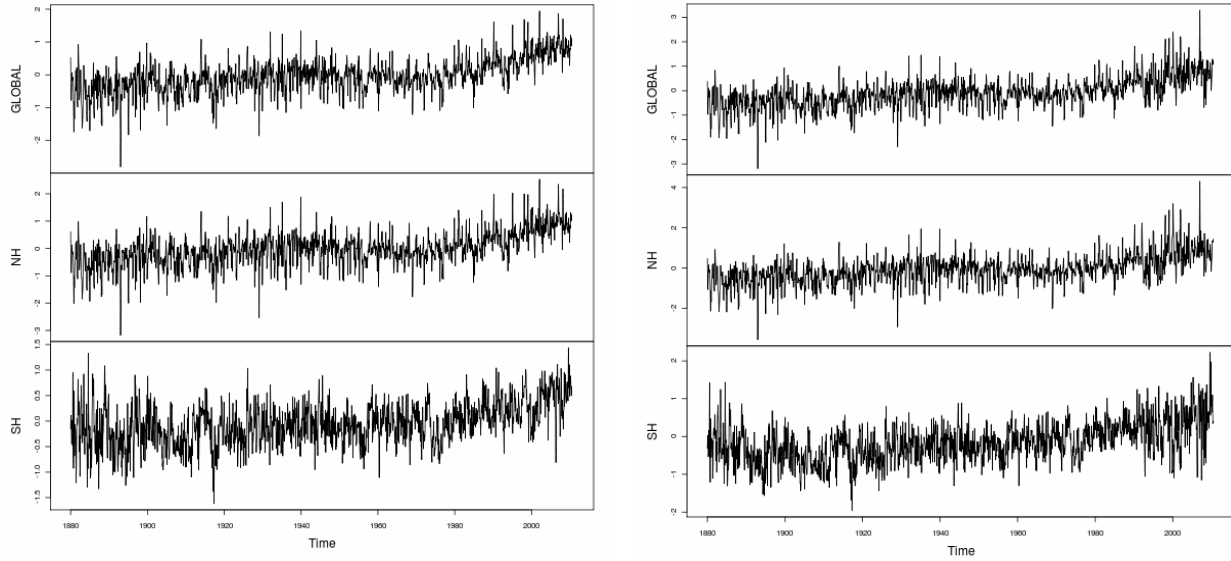


Figure 1-9: Left panel: Global and hemispheric averages using GHCN raw data. Right panel: Global and hemispheric averages using GHCN adjusted data. Data source: GHCN. See Appendix for calculation details

The effect of the adjustments is not fully apparent in the above presentation. The following graph shows the monthly differences between the GHCN adjusted and unadjusted data. The quantity shown is “delta”, denoting the adjusted series minus the unadjusted series.

Note that

$$\text{Delta} = \text{Adjusted} - \text{Unadjusted}$$

or, equivalently,

$$\text{Adjusted} = \text{Unadjusted} + \text{Delta}$$

hence the graph shows the quantity added to the raw data to produce the adjusted GHCN record.

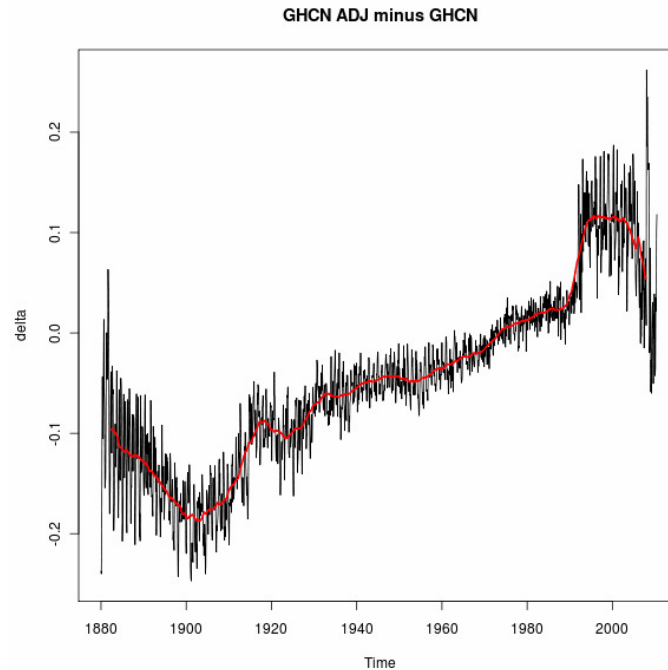


Figure 1-10: Changes (“delta”) in the global average temperature resulting from GHCN adjustments. Figure based on comparison of gridcells where both raw and adjusted data were present. data source: GHCN. See Appendix for calculation details.

There are two notable features of the graph. The first is that the adjustments trend upward. They are mainly negative prior to 1980 and positive most of the time thereafter, effectively “cooling” the early part of the record and “warming” the later record. In other words, a portion of the warming trend shown in global records derived from the GHCN-adj archive results from the adjustments, not from the underlying data.

The second is the increase in the volatility of the adjustments after 1990. The instability in the record dwarfs the size of the decadal global warming signal, with fluctuations routinely going to  $\pm 0.1$  degrees C from one year to the next.

On substantive grounds I therefore conclude that after 1990 the GHCN archive became very problematic as a basis for computing precise global average temperatures over land and comparing them to earlier decades.

#### 1.4. Summary: GHCN sampling problems.

This review of the GHCN archive leads to the following conclusions.

- GHCN is based on a compilation of 31 sources of historical temperatures as of 1997, but the monthly updates since then only track three of those sources, two of which are for US sites.
- After 1990 and again after 2005 there were massive drops in the number of GHCN sampling locations around the world. The GHCN sample for 2010 is smaller than it was for 1919.



- The drop in the size of the sample has not been uniform with respect to the type of monitoring sites: the sample has become heavily skewed towards airports, with sharp increases in the airport fraction in the past two decades.
- Nor has the declining sample been uniform with respect to latitude or altitude. In both cases, sharp step-like changes have coincided with the loss in sampling locations.
- Homogeneity adjustments intended to fix discontinuities due to equipment change or station relocations have had the effect of increasing the trend, at least up to 1990. After 1990 the homogeneity adjustments became large and seemingly chaotic, with year-by-year volatility dwarfing the magnitude of the decadal global warming signal. This points to a severe deterioration in the continuity and quality of the underlying temperature data.

## 2. Dependence of the three major global temperature products on GHCN

This section explains the reliance of all three major global temperature series (land portions) on GHCN.

### 2.1. CRU: more than 98% from GHCN

The Climatic Research Unit at the University of East Anglia produces the CRUTEM data products, which are described in Jones and New (1999), with updates CRUTEM2 (Jones and Moberg 2003) and CRUTEM3 (Brohan et al. 2006). The variance-adjusted version is denoted CRUTEM3v. The earliest version of the CRU temperature data were described in considerable detail in two reports to the US Department of Energy (Jones et al. 1985, 1986). But those data sets were superseded by CRUTEM2 and CRUTEM3. Despite promising as long ago as 2003 to release the list of input stations for the CRUTEM products, CRU Director Phil Jones has never published the exact provenance of data sources for CRUTEM (see McKittrick 2010b [48]-[62].)

In response to a Freedom of Information request in 2007<sup>3</sup>, CRU officials stated that the station data used by CRU were available from two sources: GHCN, and the US National Center for Atmospheric Research (NCAR) in the form of data sets ds540.0 and ds570.0. At the NCAR website, ds540.0 is just a mirror of GHCNv2 (<http://dss.ucar.edu/datasets/ds564.0/>). ds570.0 is the World Monthly Surface Station Climatology (<http://dss.ucar.edu/datasets/ds570.0/>), listed in Peterson and Vose (1997) Table 1 (top line) as the largest single component of the GHCNv2 archive. In other words, according to the response from CRU, their input data is identical to GHCNv2. In refusing the Freedom of Information request, CRU further stated that the GHCN includes all CRU data, or in other words CRU draws its sample from GHCN, and outside the US, the CRU sample is basically the same as GHCN:

“They [ds564.0 and ds570.0] both have a lot more data than the CRU have (in simple station number counts), but the extra are almost entirely within the USA. We have sent all our data to GHCN, so they do, in fact, possess all our data.”

In response to a subsequent request for more information under FOI regulations, the University of East Anglia Information Officer reported on April 12 2007:

<sup>3</sup> Correspondence archived online at <http://climateaudit.files.wordpress.com/2008/05/cru.correspondence.pdf>.

“I have been informed that the Climate Research Unit’s (CRU) monthly mean surface temperature dataset has been constructed principally from data available on the two websites identified in my letter of 12 March 2007 [GHCN and NCAR]. Our estimate is that more than 98% of the CRU data are on these sites.”

There is therefore no reason to assume that CRU is independent of the GHCN data base.

## **2.2. Goddard Institute of Space Studies (GISS): GHCN except Antarctica**

The global temperature data product from the Goddard Institute of Space Studies at NASA uses three input archives: GHCNv2 for the world outside USA and Antarctica, the US Historical Climatology Network (USHCN, also an NCDC product), and an archive of Antarctic stations from the Scientific Committee on Antarctic Research.<sup>4</sup> The USHCN is the largest US input to the GHCN, but the USHCN also applies its own quality control adjustments.

## **2.3. National Oceanic and Atmospheric Administration (NOAA): GHCN**

NOAA publishes a monthly global temperature anomaly record (<http://www.ncdc.noaa.gov/cmb-faq/anomalies.html>). They indicate on the NOAA website that the land record is taken from the GHCN archive, and no other sources are listed.

## **2.4. Conclusion re. dependence on GHCN**

All three major gridded global temperature anomaly products rely exclusively or nearly exclusively on the GHCN archive. Several conclusions follow.

- They are not independent as regards their input data.
- Only if their data processing methods are fundamentally independent can the three series be considered to have any independence at all. Section 4 will show that the data processing methods do not appear to change the end results by much, given the input data.
- Problems with GHCN, such as sampling discontinuities and contamination from urbanization and other forms of land use change, will therefore affect CRU, GISS, and NOAA. Decreasing quality of GHCN data over time implies decreasing quality of CRU, GISS and NOAA data products, and increased reliance on estimated adjustments to rectify climate observations.

# **3. Input data: Ocean Sources**

## **3.1. ICOADS: the major input source for all products**

All historical Sea Surface Temperature (SST) products are derived from the International Comprehensive Ocean-Atmosphere Data Set (ICOADS, <http://icoads.noaa.gov/>) or one of its predecessors. ICOADS combines about 125 million SST records from ship logs and a further 60 million readings from buoys and other sources (Woodruff et al. 2005). A large contributor to the ICOADS archive is the UK Marine Data

---

<sup>4</sup> <http://data.giss.nasa.gov/gistemp/sources/gistemp.html>.

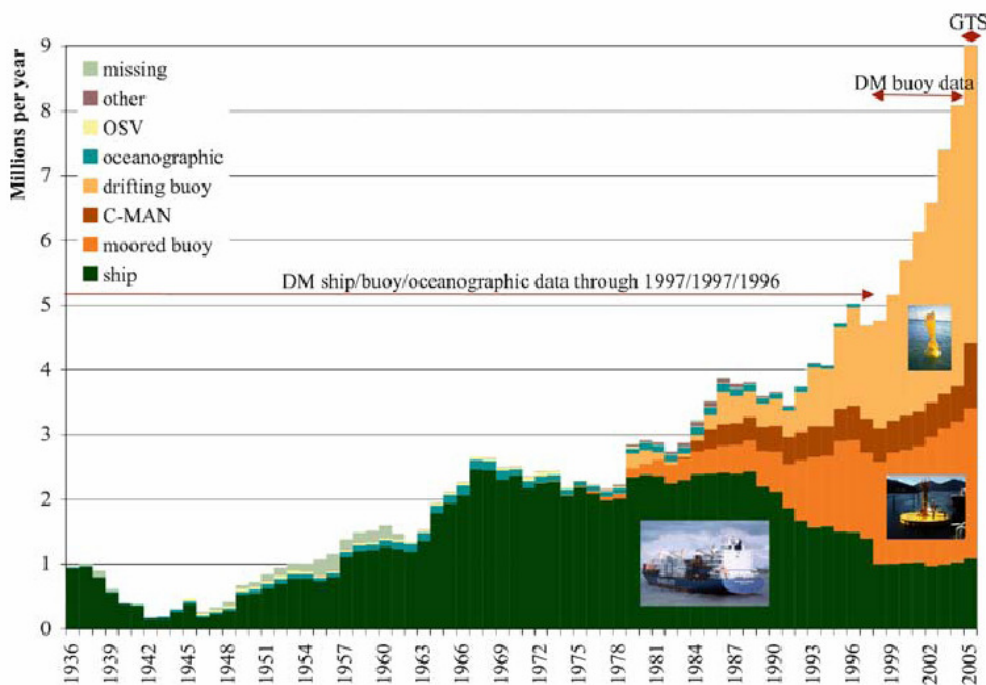
Bank. Other historical sources include navies, merchant marines, container shipping firms, buoy networks, etc.

### 3.2. ICOADS Data Sources

ICOADS draws upon a massive collection of input data, but it should be noted that there are serious problems arising from changes in spatial coverage, observational instruments and measurement times, ship size and speed, and so forth. ICOADS is, in effect, a very large collection of problematic data.

Up to the 1930s, global coverage was limited to shipping areas. This meant that most locations in the Southern Pacific region, roughly south of a straight line joining the Baja peninsula to the southern tip of Africa, had fewer than 99 observations per decade, with many areas completely blank. By the 1970s coverage was globally nearly complete, except for the oceans south of Australia, South America and Africa. Today coverage is complete except for some polar regions (Woodruff et al. 2008, see Figure 5).

The following graph is copied from Woodruff et al. (2008). It shows the changing sources of SST data from 1936 to 2005.



**Fig. 3** Annual numbers of marine reports in ICOADS, stratified by platform type for 1936–2005 (“C-MAN” refers to ODAS reports from the US land-ocean boundary zone). The indicated delayed-mode (DM) sources have been blended through 1997 (or 1996 in the case of oceanographic data) and for 1998–2004, whereas 2005 is composed exclusively of real-time data from the Global Telecommunication System (GTS). The total in 2005 is approximately 14 million (not shown), owing to many GTS drifting buoy reports, which are consolidated by the providers of the DM data (i.e., by removal of duplicates plus some compositing of fragmentary receipts) (ship photo courtesy of www.ShipPhotos.co.uk).

**Figure 3-1:** Data sources by type in ICOADS, 1936 to 2005. (taken from Woodruff 2008)

Up until 1978 the data are almost exclusively from shipping records. After 1978 the predominant source becomes drifting and moored buoys.

Ships and buoys are referred to as *in situ* measurements. Another source in recent decades has been satellite observations of the ocean surface. Rayner et al. (2003 para 58) explain the importance of these data for extending coverage outside the *in situ* zones:

At no stage in the observational record have *in situ* SSTs covered the entire ocean [Parker et al., 1995b]. In particular, the Southern Ocean has generally not been monitored. We therefore made use of satellite-based SSTs in HadISST1 to give almost complete observational coverage for recent years and a firmer basis for the [algorithms] used to interpolate the earlier, *in situ* data.

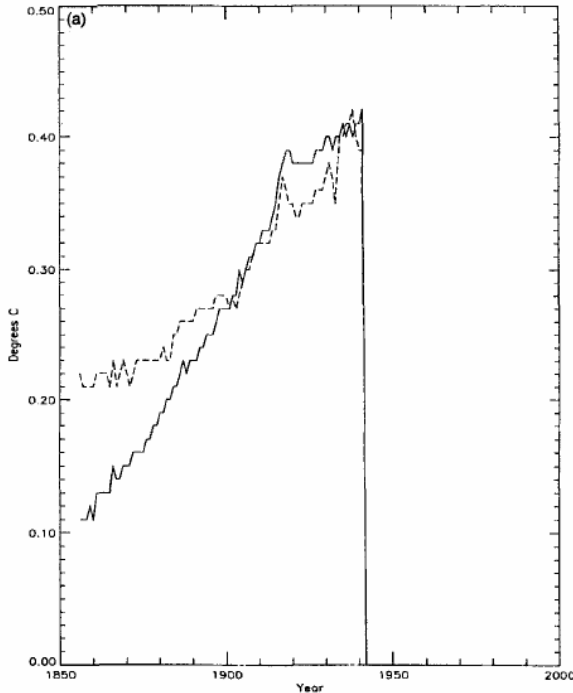
Hence the advent of satellite systems in 1978 marks an important change in the extent of the spatial sample. However, as Rayner et al. point out, satellite systems themselves have problems. Satellite measurements of sea surface temperatures become inaccurate in the presence of cloud cover and variations in atmospheric dust and aerosols. Infrared data from the Advanced Very High Resolution Radiometer (AVHRR) system can measure SST accurately but need to be calibrated to the existing SST records in order to avoid instrument bias, and they are unreliable in the presence of low cloud cover and heavy aerosol loadings. The so-called Optimal Interpolation (OI) method used by GISS employs satellite measures to interpolate SST data for complete global coverage. In the past few years, new satellite platforms (Tropical Rainfall Measuring Mission or TRMM, and the Advanced Microwave Scanning Radiometer or AMSR-E) have enabled more accurate data collection through cloud and aerosol conditions.

### **3.3. Combination of buckets and intake water**

The shipping data upon which ICOADS relied exclusively until the late 1970s, and continues to use for about 10 percent of its observations, are bedeviled by the fact that two different types of data are mixed together. The older method for measuring SST was to draw a bucket of water from the sea surface to the deck of the ship and insert a thermometer. Different kinds of buckets (wooden or Met Office-issued canvas buckets, for instance) could generate different readings, and were often biased cool relative to the actual temperature (Thompson et al. 2008).

Beginning in the 20<sup>th</sup> century, as wind-propulsion gave way to engines, readings began to come from sensors monitoring the temperature of water drawn into the engine cooling system. These readings typically have a warm bias compared to the actual SST (Thompson et al. 2008). US vessels are believed to have switched to engine intake readings fairly quickly, whereas UK ships retained the bucket approach much longer. More recently some ships have reported temperatures using hull sensors. In addition, changing ship size introduced artificial trends into ICOADS data (Kent et al. 2007).

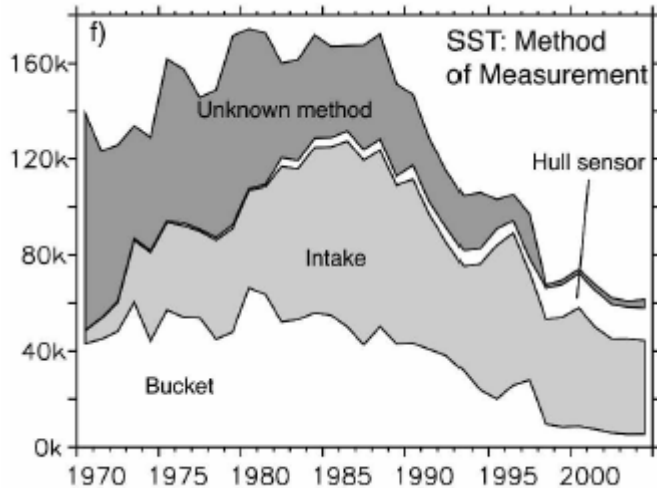
Up until recently the conventional thinking on the sources of SST data from ships was that there was an “abrupt transition from the use of uninsulated or partially insulated buckets to the use of engine inlets” in December 1941, coinciding with the entry of the US into WWII (Folland and Parker 1995). Consequently, the UK Hadley Centre adjusts the SST record using the Folland-Parker estimated factor, which ends abruptly in 1941, on the assumption that use of bucket-measurements also ended at that point.



**Figure 3-2:** From Folland and Parker (1995, Figure 19). Annual corrections to global SST (solid line); comparison to earlier proposed corrections by Bottomley (dashed line). Corrections rise from  $\sim 0.1\text{C}$  to  $\sim 0.4\text{C}$  over the 1850-1940 interval.

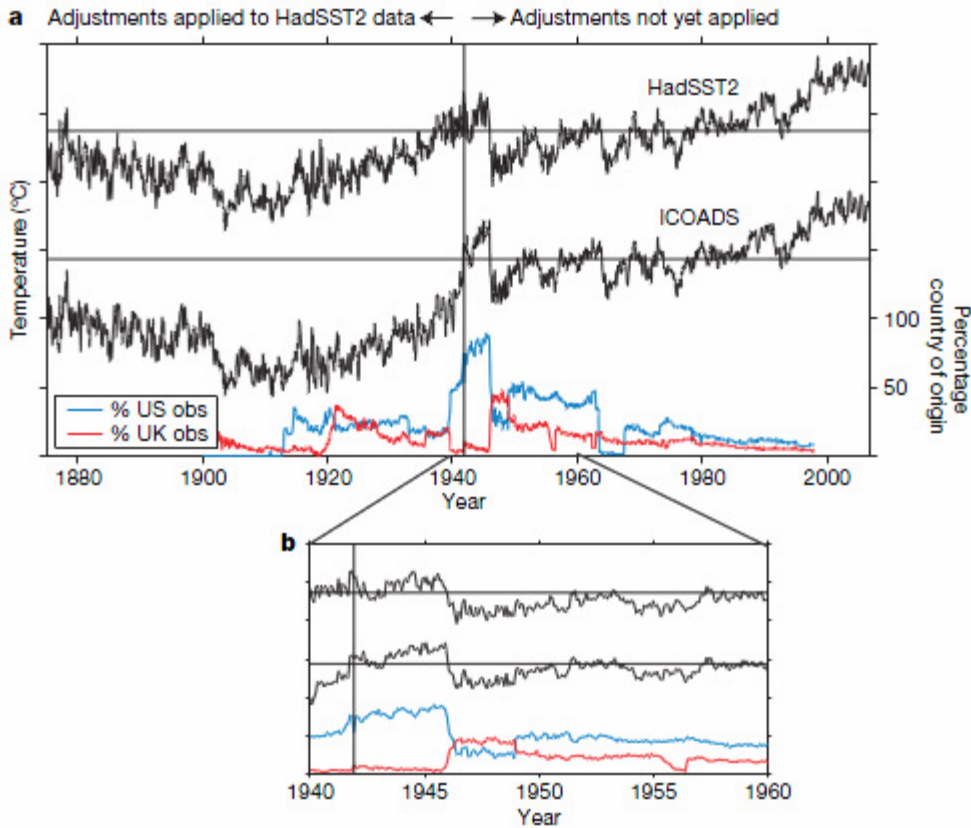
More recently, WMO ship metadata has been used to provide a clearer picture of the measurement procedures associated with ICOADS records. The following graph, taken from Kent et al. (2007), shows the changing mix of techniques used in shipping data since 1970. It is immediately apparent that the Parker-Folland analysis was incorrect to assume that bucket-measurements gave way to engine intake measurements “abruptly” in 1945, since as of 1980 they still accounted for about half the known methods for ship-derived ICOADS data.

It is also apparent that from 1970 to 1990 there was a steady increase in the fraction coming from engine intakes. As these data are biased warm relative to the actual temperature, if it had been assumed that the engine-intake fraction was constant whereas it was actually growing, it implies a possible source of upward bias to the trend over this interval.



**Figure 3-3:** SST measurement methods used on ships in the ICOADS archive. Source: Kent et al. 2007 Figure 2f.

The efforts of Kent et al. (2007) to digitize the shipping metadata had another payoff shortly thereafter. In 2008 it was reported in *Nature* magazine (Thompson et al. 2008), based on the Kent data, that a further problem with SST data had been noted at the 1945-46 transition. As with the 1941 transition, it was an odd, abrupt step in the ICOADS data, which became clearer after filtering the series and removing the effects of the El Niño and other periodic cycles. The following figure, copied from Thompson (2008, Figure 4), shows the Hadley Centre SST series (using the Folland-Parker bucket adjustment) in the top panel, the unadjusted ICOADS series in the bottom panel, and two lines near the bottom showing the Kent et al. (2007) estimate of the percentage of ICOADS data taken from, respectively, UK and US ships over the interval. From 1940 to 1945, the fraction of data coming from US ships rose sharply to over 80%, but with the end of WWII there was a jump in UK data and a drop in US data, with UK contributions going from about 0% to about 50% of the total within one year. At the same time the ICOADS average fell by about 0.5 °C.



**Figure 3-4:** ICOADS and Hadley SST data compared to fraction of record from US and UK shipping sources. Taken from Thompson et al (2008).

Thompson et al. do not recompute the global average temperature to assess the effect, though they note that the impact may be substantial in the mid-20<sup>th</sup> century. Two implications that can be drawn directly are as follows.

- If the visible discontinuity in the Hadley SST series at 1945 is resolved by raising the post-1945 data so that it becomes continuous with the pre-1945 series, then the series will become flat, implying no warming, from about 1940 to the late 1990s, substantially changing the current understanding of 20<sup>th</sup> century global warming. If, on the other hand, the discontinuity is resolved by lowering the pre-1945 series so that it joins the post-1945 series, it will imply a much larger, and unbroken, warming trend through the 20<sup>th</sup> century than previously supposed. Either way, a massive revision to the current understanding on global warming comes down to a largely arbitrary decision about how to fix an odd, recently-discovered discontinuity in a shaky data series.
- Despite the massive number of records in the ICOADS system, the overall global average is apparently very sensitive to seemingly minor changes in the sampling profile (in this case US versus British ships), implying that through much of the 20<sup>th</sup> century, the SST record lacks robustness to shifting sample sources to a greater extent than is indicated simply by counting the number of individual observations.

The climatological community appears to be uncertain about what to make of this oceanic “blip” and, possibly, a similar one in the land record. In the climategate archive, email 1254108338.txt dated September 27 2009 from Tom Wigley to Phil Jones and Ben Santer reads as follows.

From: Tom Wigley <wigley@xxxxxxxxxxx.xxx>  
 To: Phil Jones <p.jones@xxxxxxxxxxx.xxx>  
 Subject: 1940s  
 Date: Sun, 27 Sep 2009 23:25:38 -0600  
 Cc: Ben Santer <santer1@xxxxxxxxxxx.xxx>

<x-flowed>  
 Phil,

Here are some speculations on correcting SSTs to partly explain the 1940s warming blip.

If you look at the attached plot you will see that the land also shows the 1940s blip (as I'm sure you know).

So, if we could reduce the ocean blip by, say, 0.15 degC, then this would be significant for the global mean - but we'd still have to explain the land blip.

I've chosen 0.15 here deliberately. This still leaves an ocean blip, and i think one needs to have some form of ocean blip to explain the land blip (via either some common forcing, or ocean forcing land, or vice versa, or all of these). When you look at other blips, the land blips are 1.5 to 2 times (roughly) the ocean blips -- higher sensitivity plus thermal inertia effects. My 0.15 adjustment leaves things consistent with this, so you can see where I am coming from.

Removing ENSO does not affect this.

It would be good to remove at least part of the 1940s blip, but we are still left with "why the blip".

Let me go further. If you look at NH vs SH and the aerosol effect (qualitatively or with MAGICC) then with a reduced ocean blip we get continuous warming in the SH, and a cooling in the NH -- just as one would expect with mainly NH aerosols.

The other interesting thing is (as Foukal et al. note - from MAGICC) that the 1910-40 warming cannot be solar. The Sun can get at most 10% of this with Wang et al solar, less with Foukal solar. So this may well be NADW, as Sarah and I noted in 1987 (and also Schlesinger later). A reduced SST blip in the 1940s makes the 1910-40 warming larger than the SH (which it currently is not) -- but not really enough.

So ... why was the SH so cold around 1910? Another SST problem? (SH/NH data also attached.)

This stuff is in a report I am writing for EPRI, so I'd appreciate any comments you (and Ben) might have.

Tom.

Jones' reply (1254147614.txt) the next day, was, in part:

I'm told that the HadSST3 paper is fairly near to being submitted, but I've still yet to see a copy. More SST data have been added for the WW2 and WW1 periods, but according to John Kennedy they have not made much difference to these periods.

Here's the two ppts I think I showed in Boulder in June. These were from April 09, so don't know what these would look like now. SH is on the left and adjustment there seems larger, for some reason - probably just British ships there?

Maybe I'm misinterpreting what you're saying, but the adjustments won't reduce the 1940s blip but enhance it. It won't change the 1940-44 period, just raise the 10 years after Aug 45.

I expect MOHC are looking at the NH minus SH series re the aerosols. My view is that a cooler temps later in the 1950s and 1960s it is easier to explain.

In Wigley's email to Jones, he notes that if the blip is removed in the 1940s, there is continuous warming in the Southern Hemisphere, but it cannot be explained by solar changes, and it creates other problems in comparisons with the NH in the pre-1940 interval (“why was the SH so cold around 1910? Another SST



problem?"). Jones replies that the adjustments being contemplated in the Met Office-Hadley Centre (MOHC) go in the other direction than Wigley is speculating, enhancing rather than reducing the 1940s blip, and raising the data after 1945 to meet the pre-1945 series.

In other words, as of *last September*, large and arbitrary changes to global SST series were being debated in response to the Thompson et al. paper, either of which could fundamentally change the picture of mid-century warming and possibly create new discrepancies with climate models.

A further problem, similar to the land data, is a steady decline in the number of ships willing to supply data for ICOADS in recent years ([http://icoads.noaa.gov/marcdat2/MARCDAT\\_II\\_final.pdf](http://icoads.noaa.gov/marcdat2/MARCDAT_II_final.pdf)) (see Figure 3-3). While buoy data and satellite measures can help maintain data coverage, there are problems with each of these. Concerns about satellites were explained in the previous section. With regard to buoys, the new worldwide ARGO float network ([www.argo.net](http://www.argo.net)) provides, as of 2003, complete global coverage of the oceans to a depth of 2000 meters, for measurements of temperature, salinity and currents. However it does not measure SST directly, instead its profiling begins at a depth of 10 meters below sea level, and its intake pumps switch off at 8 m below sea level. ICOADS is considering making a request to the Argo system to install thermistors that would permit SST readings ([http://icoads.noaa.gov/marcdat2/MARCDAT\\_II\\_final.pdf](http://icoads.noaa.gov/marcdat2/MARCDAT_II_final.pdf)).

### 3.4. Sea Ice

SST series require estimates of areas of the ocean covered in ice. However, ice-covered regions are hazardous for shipping so data are sparse prior to the satellite era (c. 1978). Rayner et al. (2003) describe the Walsh NH sea-ice concentration charts which span 1901-1995. These are based on available shipping observations of icebound regions, however, Rayner et al. note that only the margins can be observed and coverage beyond them had to be assumed uniform. Also they note that there are no data at all for fall and winter months (September—March) from 1901 to 1956, so sea ice concentration in the marginal zones had to be estimated based on summertime data. HadISST uses satellite observations after 1978, which required an adjustment to make the series compatible.

Data on sea ice extent around Antarctica only became available via satellite observations beginning in 1973. Prior to that there were some observations based on research expeditions. HadISST uses a German record for 1929-1939, repeating it backwards in time to 1871. A Russian research record is used for the 1947-1962 interval. Other years were interpolated until satellite measures became available.

### 3.5. SST versus air temperature

When producing global records, SST is combined with GHCN data on the assumption that the two together create an average of near-surface air temperature. On the CRU FQA page (<http://www.cru.uea.ac.uk/cru/data/temperature/>) the question as to why SST are used is answered as follows (emphasis added):

Over the ocean areas the most plentiful and most consistent measurements of temperature have been taken of the sea surface. Marine air temperatures (MAT) are also taken and would, ideally, be preferable when combining with land temperatures, but they involve more complex problems with homogeneity than SSTs (Rayner et al., 2003). The problems are reduced using night only marine air temperature (NMAT) but at the expense of discarding approximately half the MAT data. **Our use of SST anomalies implies that we are tacitly assuming that the anomalies of SST are in agreement with those of MAT.** Many tests show that NMAT anomalies agree well

with SST anomalies on seasonal and longer time scales in most open ocean areas. Globally the agreement is currently very good (Rayner et al, 2003), even better than in Folland et al. (2001b). However, some regional discrepancies in open ocean trends have recently been found in the tropics (Christy et al., 2001).

Marine air temperature records are very sparse and have been affected by the growth in ship height over the century, meaning that air temperatures are not strictly comparable over time except in the cases where they are measured at a consistent height above sea level. The Christy et al. (2001) reference is interesting in this regard. Christy et. al. focused on locations where they could directly compare air and SST readings in the same places. They examined 1979-1999 SST and Marine Air Temperature (MAT) data from ships, as well as data from weather satellites, weather balloons and a network of buoys in the tropical Pacific. The buoy network data are especially useful since they measure temperatures at one meter below the surface and three meters above it in the same location. In all comparisons of SST and air temperature they found that the ocean has been warming relative to the air, indicating that SST overstated air temperature trends. Moreover, three of the air temperature data sets (satellite, balloon and reanalysis) indicate marine air temperatures just above the ocean surface had been *cooling* throughout the tropics at an average rate of between 0.01 and 0.06 °C per decade since 1979, even while the SST data showed *warming*. The authors re-calculated 1979-1999 global average temperatures over intervals where air temperature data were available instead of SST, and the global trend was reduced by 0.05 °C per decade.

The CRU claim that the SST/MAT agreement is “very good”, relies on Rayner et al. (2003). Section 5 of that paper compared the globally-averaged SST product (HADISST) to three air temperature data sets: a sparse basic set derived from ship reports and subject to some basic corrections (denoted 42N), the 42N set subject to revised adjustments for increasing average ship heights over the 20<sup>th</sup> century (denoted 43N), and a smoothed, interpolated version of 43N blended with satellite-measured SST data to obtain complete global coverage, denoted HadMAT1. When averaged up to the global level, trends through all four series were nearly identical, though the HadMAT1 case is less than compelling since it assimilates SST in its construction. In comparisons confined to locations where both air temperatures (in some cases from small islands) and SST’s were available, a distinct pattern was found in the Southern Hemisphere, in which the MAT-minus-SST difference declines by about 0.2 °C over the 20<sup>th</sup> century (Rayner et al. 2003 Figure 17). In other words SST warms relative to the corresponding air temperatures. There appears to be a similar but smaller effect in the tropics and NH (after 1980) as well. Unfortunately the information is presented in Rayner et al. (2003) in the form of small graphs in which the scale is large relative to the size of the trends, and no numerical trends are provided. In the concluding section of the paper they comment (p. 22):

[There] remains a small cooling in NMAT relative to SST in the Southern Hemisphere from the early 1990s onward, only partially ameliorated by the revised deck height corrections, which needs more investigation.

This does not do justice to the evidence presented in their Figure 17, but at least points to the fact that the problem noted by Christy et al. exists and needs further investigation.

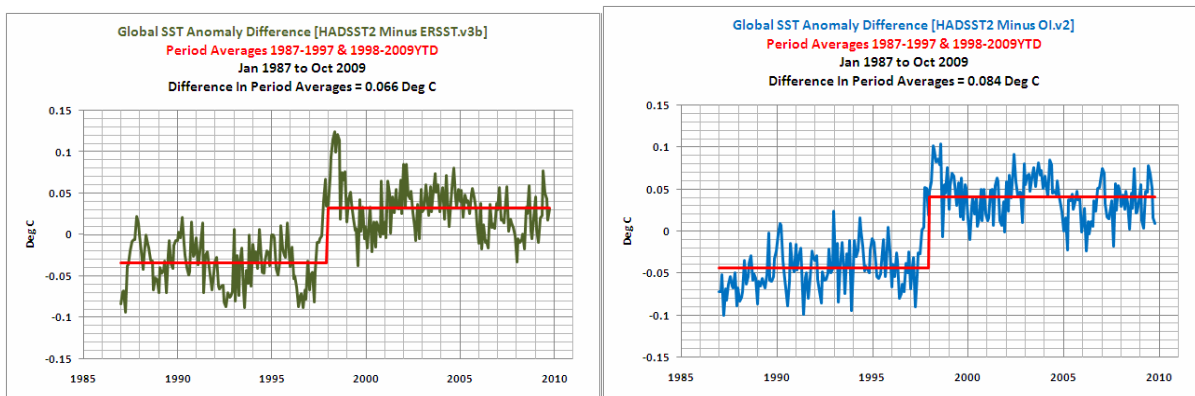
### 3.6. SST data products: Hadley Centre

The Hadley Centre produces two data sets: HADSST2 and HADISST. Descriptions are available at [www.hadobs.org](http://www.hadobs.org).

HADSST2 takes the ICOADS data, applies some homogeneity corrections, including the Folland-Parker adjustment with a slightly longer phaseout (to 1942) and forms it into 5degree grid cells, with no attempt to interpolate missing regions. The methods are presented in Rayner et al. (2006). One important feature of HADSST2 is that they changed data sources in 1998. Up to 1997 HADSST2 used the ICOADS data, then as of 1998 it uses a subset of ICOADS called the Near Real-Time (NRT) Marine Observations system (<http://icoads.noaa.gov/nrt.html>). The ICOADS web site has a brief, and somewhat cryptic, caution about combining the two data sets (emphasis added):

Near-real-time statistics based on NCEP GTS (limited products, **not fully consistent with ICOADS**; production scheduled to end in January 2011)  
(<http://icoads.noaa.gov/products.html>)

This prompted blogger Bob Tisdale to compare HadSST2 against the other SST products to see if there is a discrete change at 1998. The following graphs show the difference between HADSST2 and the corresponding series used by NOAA (ERSST.v3b) and GISS (OI.v2). NOAA uses ICOADS throughout; GISS also switches to an automated system at 1998 but does not exhibit a step-change. A relative step-change in HADSST2 coinciding with the switch to the NRT system is apparent:



**Figure 3-5:** HADSST2 minus ERSST.v3b (left) and OI.v2 SST series (right).

source: <http://bobtisdale.blogspot.com/2009/12/met-office-prediction-climate-could.html>

As noted, the NRT system will be phased out at the end of 2010 since the ICOADS system is now sufficiently automated to allow continuous updating, so Hadley will likely switch back to the ICOADS source. It will be interesting to see if there is a discrete drop in the HADSST2 series as of the start of 2011.

HADISST provides globally “complete” coverage, or in other words, numbers for every grid cell, using one of two interpolation methods. The main data source is the UK Met Office’s Marine Data Bank, supplemented with ICOADS data up to 1995. The Folland-Parker adjustment is applied up to 1941. Hadley uses a numerical method based on principal components analysis for infilling missing grids cells. After 1982 they combine the SST observations with satellite data in the interpolation algorithm.

### 3.7. NOAA

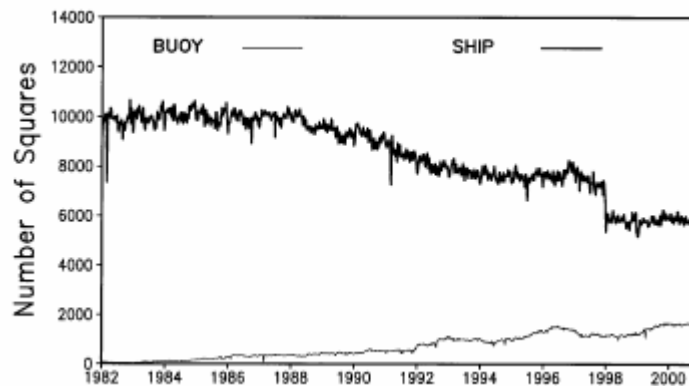
NOAA produces the Extended Reconstruction Sea Surface Temperature, or ERSST. The current version is v3b. It is based on ICOADS data. Initially NOAA used AVHRR satellite observations after 1985 to improve coverage in polar areas. This edition was called ERSST v3. However they noted that it reduced the trend slightly and deemed this effect a cold bias, so the satellite data were removed for version v3b.

<http://www.ncdc.noaa.gov/oa/climate/research/sst/ersstv3.php>

### 3.8. GISS

GISS uses another NOAA product, the Reynolds et al. (2008) Optimal Interpolation version 2 (OI.v2) data base. This is based on ICOADS up to 1998. Thereafter, like Hadley, they switch to a subset that are continuously updated. The updated subset is weighted towards buoy data since many shipping records are provided in hard copy. OI.v2 also uses AVHRR satellite retrievals to improve the interpolation for unsampled regions. Unlike the ERSST data set the satellite input is still used in OI.v2.

The following figure shows the drop in sampling points at 1998 in the OI.v2 record



**Figure 3-6:** Ship and buoy samples used in NOAA OI.v2.

Source : Reynolds et al. Figure 2.

The step down in ship records at 1998 is clearly visible. Information about the changed spatial distribution of the sample is not given. Note that ship observations as of 1998 were primarily engine intake-derived. (see Figure 3-3), which have a warm bias relative to buoys (Thompson et al. 2008). Reynolds et al. (2008) calculated a ship-minus-buoy difference of 0.13 °C in their data, but do not apply a correction because they found their calculations too uncertain to identify the appropriate adjustment (p. 1615).

Further information is at [http://www.emc.ncep.noaa.gov/research/cmb/sst\\_analysis/](http://www.emc.ncep.noaa.gov/research/cmb/sst_analysis/)

### 3.9. Conclusions about SST data

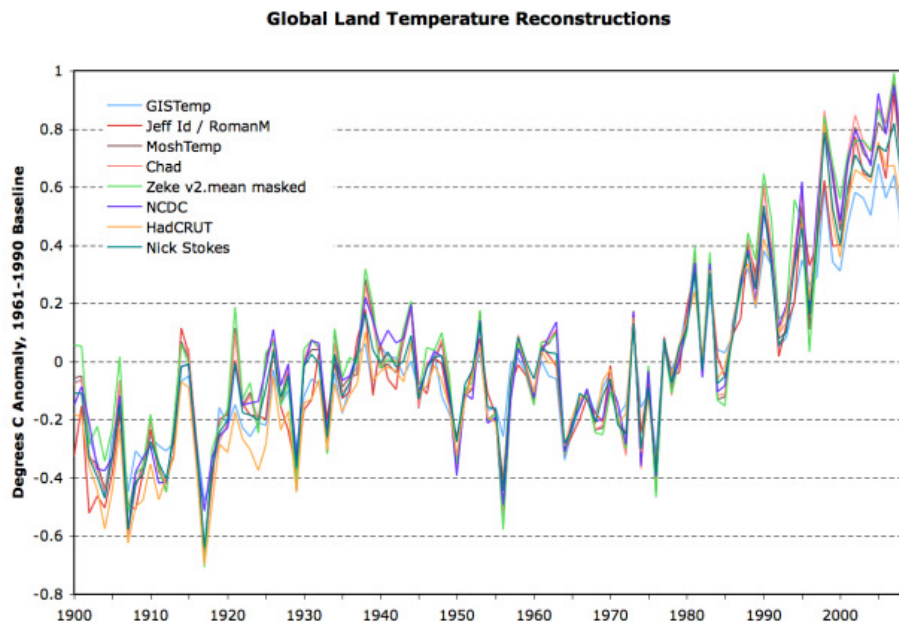
- The underlying data are of very questionable quality prior to 1950 or so. Discrepancies among sources include types of buckets, use of buckets versus engine intakes, measurement conditions, ship height, etc., all of which require large, uncertain and somewhat arbitrary adjustments.
- As recently as last fall, it has been realized that major revisions to the 20<sup>th</sup> century record may still be necessary due to discontinuities in the mid-century ship-derived data.

- The use of SST trends as a measure of air temperature trends has not been securely established, and there is evidence that SST trends exaggerate air temperature trends in the tropics and SH.
- The major compilations rely heavily on numerical methods to infill missing regions, and different teams take conflicting views on the value of using satellite retrievals to provide measurements in sparsely-sampled areas.
- The 1998 switch to a Near-Real Time data set by Hadley appears to have coincided with a step-change upwards in their series relative to other series.

## 4. A Brief Description of Processing Steps

Over the past year there have been efforts by a group of independent bloggers, as well as the Muir Russell Review Team (<http://www.cce-review.org/>), to independently produce globally-averaged land temperature anomaly series from GHCN. The Muir Russell team noted that although they had no prior experience in the area it only took them about two days to download the data and produce a plausible set of results. The efforts of bloggers are noteworthy since many of them are engineers or computer scientists with advanced skills in data extraction and coding, so their efforts have led to considerable clarification of the procedures.

The following chart from the website “The Air Vent” shows the comparison of results for the global land average from various bloggers as well as the traditional big three sources.



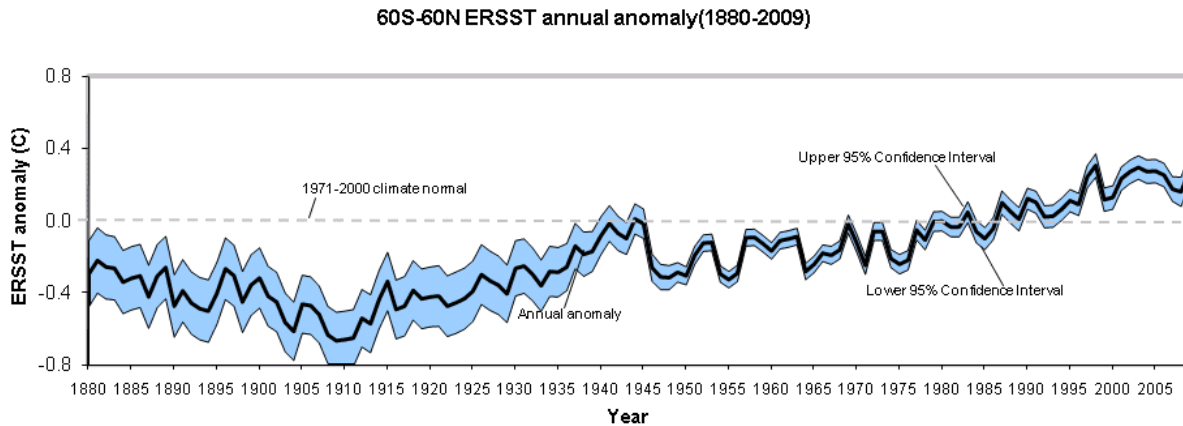
**Figure 4-1:** Blogger replications of Globally-averaged Land Temperature Anomalies.

Source: <http://noconsensus.wordpress.com/2010/07/14/gridded-global-temperature/>

There are clear similarities, although in the ending segment the spread widens to about 0.4 C. Many of the bloggers use a variation on the NOAA method to estimate the baseline from which anomalies are

computed. The similarities across all methods imply that *given the input data, the subsequent processing steps do not make much difference to the global average.*

It is interesting to compare the land records to the oceanic average (taken from ERSST v3b.)



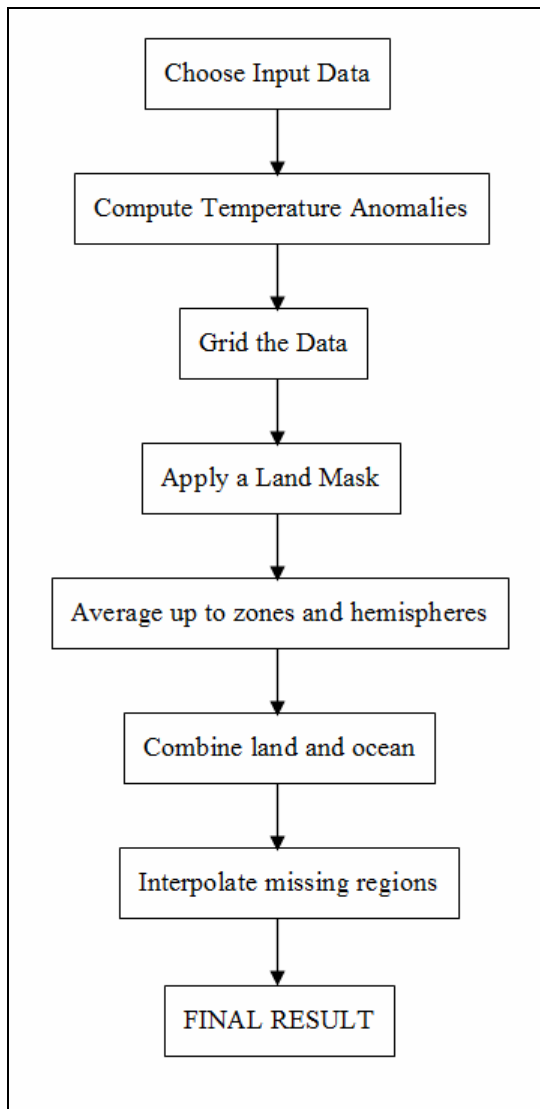
**Figure 4-2:** Average ocean temperature anomaly, 1880-2009. Source: <http://www.ncdc.noaa.gov/oa/climate/research/sst/ersstv3.php>

Note the difference in vertical scales. Over the post-1980 interval the SST record goes from about zero to about 0.3 °C. The land record goes from about zero to about 0.8 °C. *Clearly the quality of the land record after 1980 is a crucial element in determining the extent of global warming in recent decades.*

Another interesting aspect of Figure 4-2 is the so-called 95% Confidence Interval. Compare it to Figure 3-4. The ICOADS data drops outside the lower bound in the early 20<sup>th</sup> century, a shift attributed by Folland and Parker to inaccuracies resulting from reliance on bucket measurements. Then in 1945, the temperature mean drops about 0.4 °C, an effect also considered a spurious artifact of measurement irregularities (Thompson et al. 2008). In both cases, spurious effects due to measurement error cause jumps and changes that fall outside the 95% Confidence Interval. In other words, the “official” uncertainty estimates do not encompass measurement errors already known to exist. Taking the 1946 value in Figure 4-2 as an example, NOAA claims that it is 95 % confident that the SST value falls in the blue band, that is, between approximately -0.3 °C and -0.5 °C. Yet it is already known that the necessary adjustments to correct for the effect of UK ships re-entering the sample after WWII will be outside those bounds. The conclusion is that a graph like Figure 4-2, which is taken from the NOAA web site, fails to disclose the actual magnitudes of the uncertainties in the record arising from all causes, including measurement discontinuities and other sampling problems.

#### 4.1. Flowchart

The following diagram sets out the major operations involved in producing the global average temperature series.



**Figure 4-3:** Steps in producing global average temperature series.

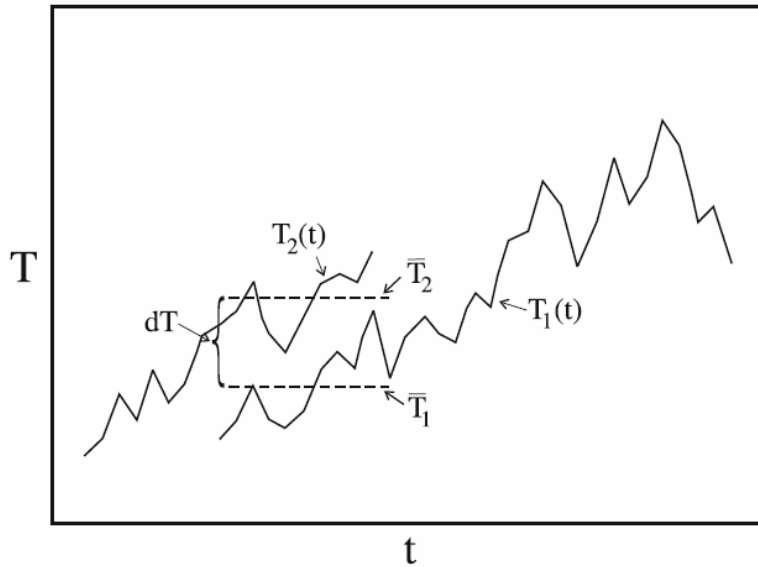
## 4.2. Notes on individual methods: GISS, CRU and NOAA

All temperature products rely on GHCN (land ) and ICOADS (ocean). Variations on these are often more apparent than real. For instance, CRU uses ds570.0 at NCAR, also known as the World Monthly Surface Station Climatology (WMSSC). But it is a subset of GHCN. GISS uses the US Historical Climatology Network in the US, but this is also a subset of GHCN, subject to some additional adjustments not applied in the GHCN methods.

### 4.2.1. GISS

GISS combines the data into grid cells and then computes anomalies. The GHCN archive needs to be reduced so that multiple records from the same place are combined. Since records may have unexplained offsets from one another, and may not cover the same time periods, this cannot be done by simple

averaging. Figure 4-4 (a copy of Figure 2 from Hansen et al. 1999) shows a schematic example.  $T_1$  is the longer series.  $T_2$  is a shorter series believed to be from the same location. If they were simply averaged together the result would contain an abrupt shift during the overlap period. Therefore  $T_2$  needs to be shifted down to line up with  $T_1$ . The size of the shift,  $dT$ , is called the “offset”.



**Figure 4-4:** Computing offsets for series to be combined. Taken from Hansen et al. (1999).

The GISS method for computing offsets involves selecting one reference station and then lining another station up to conform with it. Once the first two are combined, subsequent stations from the same location can also be added in. GISS will add in other stations without overlap as well, as long as there is no more than 10 years' gap between the single record and the reference record, and the mean temperatures for the adjacent 5-year periods differ by no more than one standard deviation (Hansen et al 1999). GISS only retains site records that are at least 20 years long.

GISS anomalies are calculated after gridding. They take the centerpoint of the grid and weight each record within the grid on a declining basis, the further it is from the center. Then they compute the 1951-1980 mean for the grid cell and subtract it to produce anomalies (Hansen et al. 1999).

GISS uses two different compilations of SST by Reynolds and Smith, one from 1950 to 1981 and one after 1982. Each of these uses numerical methods to extend coverage to the entire ocean, the former using principal component methods and the latter using satellite records.

Over land, GISS applies a processing step to correct for urbanization that involves constructing wide-area averages and then forcing nearby site records to follow them (see next section). Since ice-covered areas are omitted in GISS, this results in extrapolating land values over ice-covered ocean regions, especially in the Arctic.



#### **4.2.2. CRU**

CRU combines obviously duplicated (identical) records, then computes anomalies, then grids the resulting series (Brohan et al. 2006). Anomalies are computed by subtracting the 1961-1990 mean (the so-called “normal”). If the record does not contain at least 15 of the 30 years from 1961 to 1990 it is deemed not to have enough data to estimate the normal. In that case the normal is drawn either from WMO records if possible or by using surrounding station values. Then the anomaly series inside a 5 degree gridcell are simply averaged together, dropping any values that are more than five standard deviations from the gridcell average. Missing data are not interpolated.

#### **4.2.3. NOAA**

NOAA, like CRU, takes anomalies first, then grids. For land data, NOAA uses the GHCN-adjusted data, computes 1961-1990 anomalies of each station and then averages them together into 5 degree gridcells. For SST, 1961-1990 is again used as the anomaly base period. Then the blended land and sea series is re-set to a 1901-2000 base, so anomalies are with respect to the entire 20<sup>th</sup> century average. NOAA estimates the percentage land area in each grid cell and then combines the station record and SST record accordingly. Other products consider the cell to be all-land or all sea. Ice-covered SST areas are treated as missing, and only the land fraction is used.

#### **4.2.4. Remaining steps**

Once the grid cells for land and sea have been computed, they are averaged up into zonal bands and/or hemispheric averages, as well as into the global average.

#### **4.2.5. Conclusions: Processing steps yield minor differences**

The Muir Russell Inquiry sought to demonstrate that the CRU global land-based record could be easily replicated. They made no effort to replicate the minute details of processing steps by different teams, instead they just downloaded the data and conducted some straightforward averaging, yet they produced a global average temperature series with a strong resemblance to the CRU series. Their conclusion was that anyone who wants to replicate the major features of the CRU global average temperature anomaly series can do so. But there is another, somewhat more significant implication. *The processing steps make little difference, once the source data (GHCN and ICOADS) have been chosen. Quality limitations in those data sets put boundaries on the quality of any resulting global average.*

### **4.3. Conclusions about independence of data products**

Taking into account the reliance of land records on GHCN and the reliance of SST records on ICOADS, there is relatively little true measurement independence across the different global temperature products. The different teams do not introduce any major differences as a result of their processing steps, although there are small but important differences in some decades.

Overall, the similarity of the different data products does not provide independent replication for any of them, since they all fundamentally analyse the same underlying data.

### **4.4. Conclusions so far about construction of surface temperature record**

**NON-INDEPENDENCE:**

Given overlap of input data, and minor differences attributable to processing steps, the three series (CRU, GISS and NOAA) cannot be viewed as independent. If any one is flawed because of poor quality raw data, they will all share those flaws

**MAJOR SAMPLING PROBLEMS AFTER 1980 IN THE LAND RECORD:**

The dominant warming comes from the land surface record. However coverage has fallen radically in recent decades, with the sample being skewed towards airports, low-altitude and lower latitude regions. Adjustments to GHCN to try and correct some of the major data problems become large and chaotic after 1990. Questions about the basic quality of the raw data will be discussed further in the next Section.

**UNRESOLVED CONCERNS ABOUT SST RECORD:**

The underlying data are of poor quality prior to 1950 or so due to reliance on shipping records. Data are combined from very different sources, with discrepancies among instruments, techniques, measurement conditions etc. requiring large and sometimes arbitrary adjustments. The use of SST as a measure of air temperature has been shown to be a potential source of upward bias. As recently as last fall, major flaws in the mid-20<sup>th</sup> century record have been discovered based on new examinations of ship-derived data. The major compilations rely heavily on numerical methods to infill missing regions, and different teams take conflicting views on the value of using satellite retrievals to provide measurements in sparsely-sampled areas.

## 5. Quality of the raw GHCN data

### 5.1. Biases related to urbanization and land use change

Up to this point the discussion has focused on the changing numbers and locations of surface records. This section looks at the question of possible biases in the air temperature record due to urbanization and other types of land-use change.

#### 5.1.1. General acknowledgment of problem; claims that it has been rectified

Climatic data are processed versions of temperature records. Temperatures at land-based observational sites can be affected by modifications to the local land surface due to deforestation, introduction of agriculture, road-building and urbanization, as well as changes in monitoring equipment, measurement discontinuities, and so forth; and by local emissions of particulates and other air pollutants. These are *non-climatic* influences, since they are driven by local, rather than global, climatic forcing. Hence the raw temperature record must be adjusted, if possible, to remove the effects of these things and reveal the *climatic* record. An ideal record of surface climatic changes would require a monitoring site untouched by human development, the equipment for which was consistent and perfectly maintained over the entire measurement interval. However, as shown in the previous section, GHCN data almost never satisfies these ideals. The GHCN adjustments are intended to remedy problems of equipment discontinuity, at least in cases where they are large enough to be detected by the algorithms that look for unusual step-changes. However, other data contamination problems need to be addressed by the users of GHCN data.

As an illustration of the wide acknowledgment of the problem, the CRU web page (<http://www.cru.uea.ac.uk/cru/data/hrg/>) presents two data compilations: CRU TS and CRUTEM. The TS series are not subject to adjustments for non-climatic influences, and for that reason users are cautioned not to use them for climate analysis (see <http://www.cru.uea.ac.uk/cru/data/hrg/timm/grid/ts-advice.html>). There is an online list of FAQ's for this product.<sup>5</sup> The very first question, and its answer, are reproduced (in part) below.

### Question One

**Q1.** Is it legitimate to use CRU TS 2.0 to 'detect anthropogenic climate change' (IPCC language)?

**A1.** No. CRU TS 2.0 is specifically *not* designed for climate change detection or attribution in the classic IPCC sense. The classic IPCC detection issue deals with the distinctly anthropogenic climate changes we are already experiencing. Therefore it is necessary, for IPCC detection to work, to remove all influences of urban development or land use change on the station data....If you want to examine the detection of anthropogenic climate change, we recommend that you use the [Jones](#) temperature data-set. This is on a coarser (5 degree) grid, but it is optimised for the reliable detection of anthropogenic trends.

The implication is that the Jones data (CRUTEM) has been adjusted “for the reliable detection of anthropogenic trends.” The assumption that the adjustments are adequate is widely held. For example, Jun et al. (2008) used surface climate data in a study that tested some properties of climate models. While they are aware of the many faults of the underlying data, they dispensed with them as follows (p. 935):

Inhomogeneities in the data arise mainly due to changes in instruments, exposure, station location (elevation, position), ship height, observation time, urbanization effects, and the method used to calculate averages. However, these effects are all well understood and taken into account in the construction of the data set.

In its 4<sup>th</sup> Assessment Report, as in the previous three, the IPCC also acknowledged the problems of contaminating influences on temperature data but claimed that adjustments have been applied to remove them. This forms an essential assumption behind all the key IPCC conclusions. Global temperature trends were presented in Table 3.2 on page 243 of (Working Group I). The accompanying text (page 242) states that the data uncertainties “take into account” biases due to urbanization. The Executive Summary to the chapter (page 237) asserts that “...the very real but local [urbanization] effects are avoided or accounted for in the data sets used.” The influential Summary for Policymakers stated:

“Urban heat island effects are real but local, and have a negligible influence (less than 0.006°C per decade over land and zero over the oceans) on these values.”

The 0.006°C is referenced back to Brohan et al. (2006), where it is merely an assumption about the standard error, not the size of the trend bias itself. IPCC Chapter 9 provides the summary of evidence attributing warming to greenhouse gases. The IPCC sets aside the problem of surface data contamination as follows p. 693):

Systematic instrumental errors, such as changes in measurement practices or urbanisation, could be more important, especially earlier in the record (Chapter 3), although these errors are

<sup>5</sup> <http://www.cru.uea.ac.uk/cru/data/hrg/timm/grid/ts-advice.html>

calculated to be relatively small at large spatial scales. Urbanisation effects appear to have negligible effects on continental and hemispheric average temperatures (Chapter 3).

The citation to IPCC Chapter 3 is uninformative. That chapter does not describe the data adjustments and only briefly mentions a few studies that have found evidence of urbanization and land-use-related biases in climate data. These studies are raised only to be dismissed without much, if any, consideration of their merits.

### 5.1.2. Adjustment methods: CRU

CRU relies on subjective, and largely undocumented, methods to remove non-climatic patterns in the temperature data. Brohan et al. (2006) does not explain the actual adjustments to CRU data, it mainly focuses on the claim that any biases are very small. Brohan et al. Section 2.3 says of the temperature data they use that adjustments were made to raw temperature series but in most cases the originals were not retained, so it is now impossible to say how large the adjustments were.<sup>6</sup> This remark appears to refer to adjustments made to the GHCN raw archive, not to the subsequent changes leading to the GHCN adjusted archive. Brohan et al. assume any inhomogeneity uncertainties are symmetric around zero (p. 6). In their Section 2.3.3 they state that to properly adjust the data for urbanization bias would require a global comparison of urban versus rural records, but classifying records in this way is not possible since “no such complete meta-data are available” (p. 11). The authors instead invoke the assumption that the bias is no larger than 0.006 degrees per century.

Jones and Moberg (2003) likewise offers little information about the data adjustments. They discuss combining multiple site records into a single series, but do not discuss removing non-climatic contamination. Moreover, like Brohan et al., the article points out (page 208) that it is difficult to say what homogeneity adjustments have been applied to the raw data since the original sources do not always include this information. They emphasize that non-climatic influences must be corrected (Section 2, p. 174) for the data to be useful for climatic research. But the part of the paper that outlines the adjustments consists of only three paragraphs in Section 2.1, none of which explains the procedures. The only explanatory statement is the following (page 174):

“All 2000+ station time series used have been assessed for homogeneity by subjective interstation comparisons performed on a local basis. Many stations were adjusted and some omitted because of anomalous warming trends and/or numerous nonclimatic jumps (complete details are given by Jones et al. [1985, 1986c]).”

The two reports cited (“Jones et al. 1985, 1986c”) are the technical reports submitted to the US Department of Energy as mentioned in Section 2.1. They only cover data sets ending in the early 1980s, whereas the data typically under dispute now is the post-1979 interval. Even if the adjustments were adequate in the pre-1980 interval it is likely impossible to have estimated empirical adjustments in the early 1980s that would apply to changes in socioeconomic patterns that did not occur until the 1990s and after.

---

<sup>6</sup> “For some stations both the adjusted and unadjusted time-series are archived at CRU and so the adjustments that have been made are known [Jones et al., 1985, Jones et al., 1986, Vincent & Gullet, 1999], but for most stations only a single series is archived, so any adjustments that might have been made (e.g. by National Met. services or individual scientists) are unknown.” Brohan et al. 2006 p. 6.

In sum, the CRU cautions that its unadjusted temperature data products (TS) are inappropriate for climatic analysis, and refers users to the CRUTEM products. Yet the accompanying documentation does not appear to explain the adjustments made or the grounds for claiming the CRUTEM products are reliable for climate research purposes.

### 5.1.3. Adjustment methods: GISS

GISS uses the GHCN v2 raw (unadjusted) data (Hansen et al. 2001). Outside the US, GISS uses the population data in the GHCN archive to estimate an urbanization effect, although their method is not intended to remove temperature changes due to local aerosols or land use effects. Based on population data a station is identified as rural, small town or urban. Rural stations are identified as those with less than 10,000 population, except in the US, southern Canada and Mexico, where rural stations are identified as “unlit” locations using nighttime satellite images.

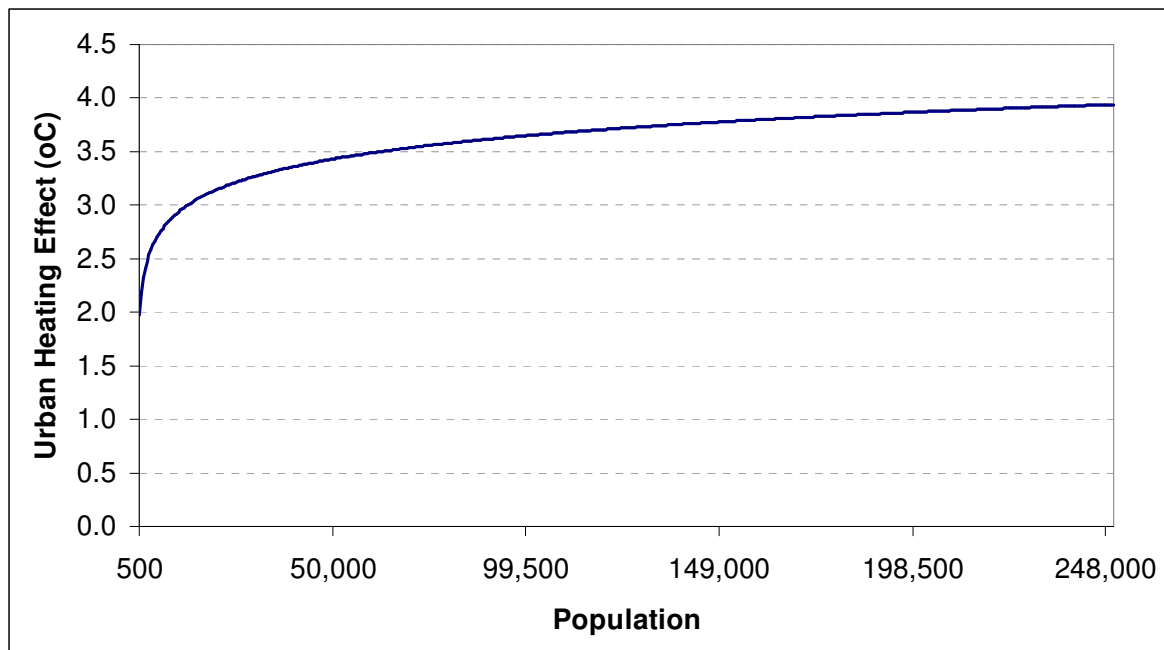
For an urban or small town station, a radius of 500 km is drawn around it and any rural stations in that distance are identified. As long as there are at least 3 such stations, the mean temperature trend is computed, and then the urban station at the center of the circle is adjusted so that the trend is the same as the rural trend. If three stations are not available within 500 km, the radius is expanded to 1000 km.

The validity of this adjustment depends strongly on the assumption that the stations identified as “rural” show the true regional climatic signal. While intuitively plausible, there are two reasons why this assumption may not be safe.

First, biases due to urbanization have long been believed to grow most quickly in the early stages of population growth. Oke (1973) showed that even small towns have measurable heating compared to the nearby rural countryside. Oke estimated that the Urban Heat Island effect (in °C) increases according to the formula

$$UHI = 0.73 \log_{10} (pop)$$

where *pop* denotes population. A graph of this relation, on population numbers from 0 to 250,000, is as follows.



**Figure 5-1:** The Oke (1973) model of the Urban Heat Island Effect. The theoretical starting value is zero if population equals one person.

This means that if a town grew from 500 to 1,000 persons over the time period of the calculation, it would be considered “rural” in the GISS analysis, yet would record 0.22 °C false warming. By contrast a town that grew from 240,000 to 250,000 persons would only add 0.014 °C false warming, yet it would be adjusted to have the same trend as the small town, which in this case would entail an increase in its warming rate.

With regard to CRU data, Brohan et al. (2006, p. 11) emphasized that reasonable urbanization adjustments should always yield reductions in temperatures: “recent temperatures may be too high due to urbanisation, but they will not be too low.” Yet the GISS method routinely yields adjustments that increase the estimated trend. Hansen et al. (2001) report that “the homogeneity adjustment changes the urban record to a cooler trend in only 58% of the cases, while it yields a warmer trend in the other 42% of the urban stations.” (p. 5).

The second problem with the assumption behind this method is that weather stations in rural areas are typically related to agriculture, and modification of the landscape to support agriculture can itself induce a warm bias. Chagnon (1999) described a temperature series collected from 1889 to 1952 at an Illinois agricultural research station from a thermometer placed in a glass tube and lowered 3 feet into the soil. The tube was relocated twice to maintain its placement in totally rural fields as the nearby university campus expanded. Air temperatures were also collected above ground at a station that was gradually encroached upon by the university campus, as well as in two nearby towns. By using above-ground stations, the USHCN estimates a warming of 0.6 °C over the interval, and deems it to be free of urbanization bias. Chagnon, however, found that the soil temperatures increased by only 0.4 °C, leading him to conclude that even small-town and university campus sites can have larger urban heating effects than are typically assumed.

Christy et al. (2006) described the differential trends in California between a non-agricultural mountain district and a nearby (valley) agricultural district, documenting a substantial warming in the agricultural district that was not detected in the mountains. He cautioned that agricultural districts may have warming biases of their own. This is an important finding in light of the migration of GHCN records to lower and lower altitudes, in other words away from mountain sides and down into valleys, where agriculture tends to be located (see Figure 1-8).

Another concern about the GISS method is that it requires adjacent rural-urban pairs, and as such is primarily applicable in the US. While 74 percent of US stations have enough nearby rural data to support estimation of an adjustment, only 37 percent of stations outside the US do.<sup>7</sup> And since more than half the non-US GHCN data come from airports, this implies a large fraction of the GISS data comes from unadjusted airport records outside the US.

Focusing on the US, where most stations are adjusted and most adjustments reduce the trend, Hansen et al. (2001) report the overall effect of the adjustments is to reduce the 20<sup>th</sup> century US warming trend by about 0.15 °C. Their estimated warming from 1900 to 1999 is 0.51 °C globally, compared to 0.32 °C in the US. Again it is noteworthy that where the adjustments are most feasible and sampling is most dense, the trend is considerably lower.

#### **5.1.4. Adjustment methods: NOAA**

NOAA uses the GHCN adjusted data (see Section 1.1.2). They do not make further adjustments for urbanization bias, though they add a constant to the standard deviation to expand the 95% Confidence Interval to indicate the additional uncertainty created by the problem (Smith and Reynolds 2005 p. 2035).

### **5.2. Ex post methods for testing the adequacy of the adjustments**

The fact that adjustments have been applied does not prove that they are adequate for the purpose. By way of analogy, suppose a cleanser is developed that is supposed to be able to kill all the bacteria on a tabletop. The effectiveness of the cleanser could not be proven simply by listing its ingredients. The proper test would be to examine an infected surface after using the cleanser, to test for the remaining presence of live bacteria. In other words, an *ex ante* test (listing the ingredients) is not enough, instead we require an *ex post* test (the absence of bacteria on the surface after using the cleanser) to prove its effectiveness.

Likewise, in the discussion of measures to remove urbanization contamination of the temperature data, the discussion has been heavily focused on *ex ante* tests, namely listing the “ingredients” of data adjustments. There has been relatively less attention paid to *ex post* tests: indeed there isn’t even any agreement about what the proper *ex post* test should be.

#### **5.2.1. Observations vs reanalysis data**

Kalnay and Cai (2003) compared surface temperature data to “reanalysis” data taken from 6-hour ahead weather forecasts archived at NCAR. The reanalysis data contains surface temperature forecasts

---

<sup>7</sup> Source: tabulation at <http://climateaudit.org/2008/03/01/positive-and-negative-urban-adjustments/>.

generated using atmospheric (but not surface) observations of wind and temperature derived from weather balloons and satellite records. Since the reanalysis data do not use direct surface measurements, Kalnay and Cai argued that they provide an approximation to a surface record free of contaminating influences from land use change. In favour of this interpretation is the observation that reanalysis data shows no difference in trends between rural and urban locations. They found a century-scale difference in minimum temperature trends between surface and reanalysis data of +0.27 C, implying a false warming much larger than had previously been estimated. They attributed this, in part, to the fact that agriculture is also a source of warming bias in temperature records, but is not picked up by traditional adjustment methods based on rural-urban comparisons.

Trenberth (2004) criticized the Kalnay and Cai methods by pointing out that land use change is not the only thing left out of the reanalysis data: it also omits information on many other potential influences on surface temperatures including changing greenhouse gas levels, volcanoes, trends in cloud cover and surface moisture changes. Kalnay and Cai (2004) responded that the omitted factors that matter, such as greenhouse gas and volcanic effects, are assimilated into the analysis via the atmospheric temperature observations. Vose et al. (2004) pointed out that the surface observations are affected by many changes in instrumentation and time of measurement. When they use data corrected for these factors (so-called HCN records, which Kalnay and Cai did not use) they get an even larger difference, implying the land use impact is twice as large as Kalnay and Cai had found. Vose et al. take this as evidence *against* Kalnay and Cai's results, on the grounds that they don't believe them.

“These estimates seem improbable and indicate to us that the NNR [reanalysis] trends are not accurate. We infer this in part because there is extensive evidence to support corrected HCN trends...We are not aware of any evidence demonstrating the reliability of the NNR surface temperature trends.”

Kalnay and Cai (2004) would have been justified in pointing out that this argument is circular. Vose et al. say they are unaware of evidence demonstrating the reliability of reanalysis data. They are aware of the Kalnay and Cai paper, of course, but they deem it not to have demonstrated the reliability of the reanalysis data since it disagrees with the HCN data, which they assume to be valid. But since Kalnay and Cai have shown the surface records to be invalid, the Vose argument requires the assumption that the Kalnay and Cai results are incorrect. In other words they conclude Kalnay and Cai are wrong on the basis of an argument that assumes Kalnay and Cai are wrong.

Simmons et al. (2004) used a different reanalysis product, ERA-40, from the European Centre for Medium-Range Weather Forecasts, and showed that it provided a reasonably good match after 1980 to the CRU data compared to the one between NCAR reanalysis data and the US data used by Kalnay and Cai. Since ERA-40 assimilates more data into its forecasting scheme than NCAR reanalysis data, they argued that the Kalnay-Cai results may simply have been due to relatively poor quality reanalysis data, especially prior to 1979. This was the position espoused by the IPCC in the AR4 (p 245):

Vose et al. (2004) showed that the adjusted station data for the region (for homogeneity issues, see Appendix 3.B.2) do not support Kalnay and Cai's conclusions. Nor are Kalnay and Cai's results reproduced in the ERA-40 reanalysis (Simmons et al., 2004). Instead, most of the changes appear related to abrupt changes in the type of data assimilated into the reanalysis, rather than to gradual changes arising from land use and urbanisation changes. Current reanalyses may be reliable for estimating trends since 1979 (Simmons et al., 2004) but are in general unsuited for estimating longer-term global trends, as discussed in Appendix 3.B.5.



The quoted paragraph is misleading in saying that the Vose et al. results “do not support” those of Kalnay and Cai: in fact they reinforce the Kalnay and Cai results; Vose et al. simply dismiss them as implausible.

There are two further salient points about the Simmons et al. paper. First, the ERA-40 reanalysis uses CRU input data as an input to its own forecast scheme (Simmons et al. paragraph [7]). Consequently, as the paper itself notes, ERA-40 data are not entirely independent of CRU data, whereas the reanalysis data in Kalnay and Cai are entirely independent of the surface data they tested it against. Second, Phil Jones of the CRU was the second author on the Simmons paper as well as Coordinating Lead Author of Chapter 3 of the IPCC Report. Consequently the fact that the IPCC endorsed the Simmons et al. argument is meaningless, since it comes down to Jones endorsing his own study. One thing he especially liked about the paper was that its results were “great” for CRU. In a 2004 email to Michael Mann (1089318616.txt) with the subject line ‘HIGHLY CONFIDENTIAL’ he wrote the following:

```
From: Phil Jones <p.jones@xxxxxxxxxxx.xxx>
To: "Michael E. Mann" <mann@xxxxxxxxxxx.xxx>
Subject: HIGHLY CONFIDENTIAL
Date: Thu Jul 8 16:30:16 2004
```

Mike,

Only have it in the pdf form. FYI ONLY - don't pass on. Relevant paras are the last 2 in section 4 on p13. As I said it is worded carefully due to Adrian knowing Eugenia for years. He knows the're wrong, but he succumbed to her almost pleading with him to tone it down as it might affect her proposals in the future !

I didn't say any of this, so be careful how you use it - if at all. Keep quiet also that you have the pdf.

The attachment is a very good paper - I've been pushing Adrian over the last weeks to get it submitted to JGR or J. Climate. The main results are great for CRU and also for ERA-40. The basic message is clear - you have to put enough surface and sonde obs into a model to produce Reanalyses. The jumps when the data input change stand out so clearly. NCEP does many odd things also around sea ice and over snow and ice.

Zhou et al. (2004) used an improved version of the NCAR reanalysis data and applied the Kalnay and Cai method to Chinese urban data, finding a warming bias of about 0.05 °C/decade.

We can summarize this section as follows.

- Kalnay and Cai argued that reanalysis data products closely emulate surface temperature conditions but are free of biases due to urbanization and land use change. They attribute the difference between observed warming and the reanalysis-computed warming in eastern US data to land use factors, and estimate a larger warm bias than previous studies had using rural-urban comparisons.
- Vose et al. found that using homogeneity-adjusted data yielded an even larger estimate of the land use effect, but claimed the Kalnay and Cai method was still wrong because they found the results implausible.
- Trenberth argued that the differences between reanalysis and surface data may reflect a number of elements missing from the reanalysis system, though Kalnay and Cai pointed out many of them are effectively included via their effects on the atmospheric temperature data.

- Simmons et al. found a smaller discrepancy between ERA-40 reanalysis data and CRU data in recent decades, and argued that incomplete data assimilation likely accounts for the trend discrepancy Kalnay and Cai found. A weakness of their argument is the overlap between input data for CRU and ERA-40.
- The use of reanalysis data as an *ex post* test for surface temperature adjustments may yet turn out to be a valid tool. The IPCC rejection of the Kalnay and Cai findings on the basis of Simmons paper is uninformative since the second author on the Simmons paper was Phil Jones, who was a Coordinating Lead Author of the relevant section of the IPCC report, and he had earlier praised the paper as being “great for CRU” – indicating a lack of disinterested objectivity.

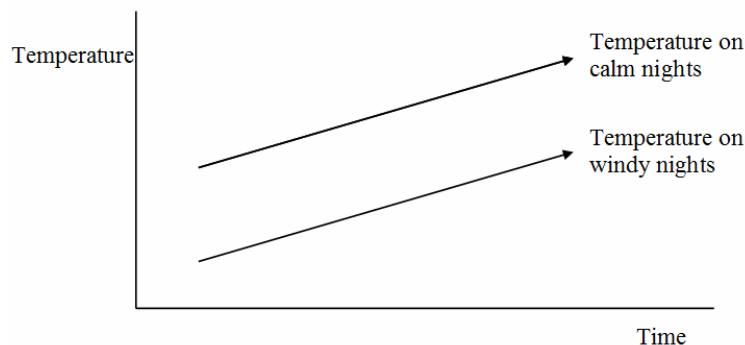
### 5.2.2. Calm vs windy conditions

UK Met Office scientist David Parker (2004, 2006) proposed an *ex post* test of the influence of urbanization on surface data based on the argument that windy conditions mitigate the urban heat island (UHI) effect. Therefore, the strength of the UHI effect could be measured by comparing temperature trends on calm nights versus windy nights. Parker did so and found that while temperature levels were lower on windy nights, the trend over time was the same as those on calm nights, indicating that the UHI did not bias the trend.

Parker’s argument rests on the premise that wind mitigates the UHI. He cited Johnson et al. (1991) for the claim that “the influence of urbanization on air temperatures is greatest on calm, cloudless nights and is reduced in windy, cloudy conditions.” What Johnson et al. said was that calm, cloudless conditions are ideal for formation of a large gradient between the temperature in an urban center versus a rural area, but they did not say they are the only conditions for creating a UHI, nor that windy conditions prevent a UHI.

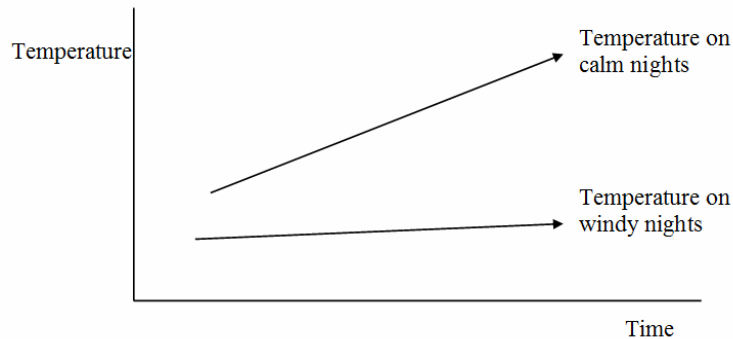
““[M]ost mid-latitude studies show that the heat island intensity (the difference between the temperature of the warmest location in the city and the background rural value) of the surface layer reaches its maximum a few hours after sunset on calm, cloudless summer nights”

(Johnson et al. 1991 p. 275). Parker’s premise is somewhat different. Schematically, his results look like the following.



**Figure 5-2:** Schematic version of Parker’s results:  
temperature trends do not differ based on windspeed.

Parker's argument is that if the UHI had a strong effect on mean temperatures, then the lines in Figure 2.2 would not be parallel, instead they would look something like this:



**Figure 5-3:** Schematic version of Parker's premise:

a strong UHI effect should generate divergent trends. Since we do not observe divergence, therefore the UHI effect is not strong.

For the premise to be true, it would have to be the case that we would expect to observe Figure 5-3 on a sample of urban sites over time, which would imply that increased urbanization was fully offset by wind. In other words, while a UHI effect clearly exists, it cannot grow (or at least it must grow more slowly) on windy nights even as urbanization grows.

The main difficulty in evaluating this line of argumentation is that Parker's argument relies on a premise not proven either by his source or by his analysis. His technique does not involve measuring the UHI effect directly, instead it assumes that wind cancels out any growth in the UHI, and then examines the effect on observed trends effect of comparing windy and calm nights. If a windy/calm trend difference is not found, he concludes there is no UHI effect on the global average. But the other alternative interpretation is that there is a UHI effect, it just grows on windy nights as well as calm nights. If it is the case that the UHI can grow even on nights with elevated wind speed, then Figure 5-2 is consistent with UHI contamination of the temperature record. The paper cited by Parker as the basis for his premise is Johnson et al. (1991). But they do not actually state that wind obliterates the UHI under all circumstances, they only state that calm conditions are ideal for forming the largest gradient between urban and rural temperatures. In other words, in terms of Figure 5-2, Johnson et al. state that the distance between the arrows will be larger, the larger is the difference in windspeed between calm and windy nights. But they do not state that the presence of a UHI effect in the data leads to a pattern like Figure 5-3.

There has been very limited examination of Parker's premise. Pielke Sr. and Matsui (2005) argued that, based on turbulent cooling mechanisms on the boundary layer (within 10 meters of the surface) we

should not necessarily expect differing trends on light and windy nights. But their analysis did not clarify what the profile should look like under the hypothesis of no UHI effect, so the matter remains unsettled.

The IPCC cited Parker's analysis in support of its conclusions that the UHI effect did not bias global trends over land. But as was the case with the debate over the Kalnay and Cai hypothesis, the IPCC's judgment is of no use in this instance, since David Parker himself was a Lead Author of the report section that dealt with his own material, and he is a regular coauthor with Phil Jones, the Coordinating Lead Author.

### 5.2.3. Socioeconomic gridding

A third hypothesis for *ex post* testing of climate data was offered separately in McKittrick and Michaels (2004) and in de Laat and Maurellis (2004). They pointed out that the spatial pattern of industrialization over land does not match the spatial pattern of warming over land in response to greenhouse gas forcing as predicted by climate models. Also, data processing methods are supposed to remove the influence of land use change, urbanization, and other factors tied to population growth and economic development. Therefore, a way to test if the homogeneity corrections are adequate is to compare the spatial pattern of trends in temperature data to the spatial pattern of measures of industrialization. If the data processing methods are adequate, there should be no correlation between warming patterns and socioeconomic patterns.

de Laat and Maurellis (2004) used industrial carbon dioxide emissions as a proxy for the extent of socioeconomic activity in a region. They showed convincingly that measured trends were higher in regions with high emissions compared to regions with low emissions, irrespective of where they set the threshold between high and low, but the gap was largest when the threshold was set fairly high. They found this difference in lower-tropospheric data as well. Their 2006 follow-up paper confirmed these results on a larger group of data sets.

McKittrick and Michaels (2004a, 2007) tested the adequacy of the data adjustments by regressing the observed 1979-2002 trends in 440 surface grid cells on a vector of climatological variables (lower tropospheric temperature trends and fixed factors such as latitude, mean air pressure and coastal proximity) augmented with a vector of socioeconomic variables, including income and population growth, Gross Domestic Product (GDP) per square km, education levels, etc. If the data have been adjusted to remove all non-climatic influences then the spatial pattern of warming trends should not vary systematically with socioeconomic indicators. But both papers rejected, at very high significance levels, independence of the surface temperature trends and the socioeconomic variables, thus concluding that the adjusted surface climatic data likely still contain residual influences of industrialization on local temperature records. They estimated that the non-climatic effects could account for between one-third and one-half of the post-1979 average warming trend over land in the temperature data.

Benestad (2004) critiqued the McKittrick and Michaels (2004a) findings on the basis of a sensitivity analysis in which he discarded all the northern hemisphere data and many of the explanatory variables, and then used the remaining data to try to predict the discarded observations. Upon finding the results weak, he argued the original findings might be spurious insofar as they cannot be reproduced on all data subsets. McKittrick and Michaels (2004b) pointed out that this was an extreme test for which no precedent exists in the literature, and the model passed more reasonable versions of such a test. McKittrick and Michaels (2007) developed a new and larger version of the data base for testing the hypothesis and found nearly identical results. Later, McKittrick and Nierenberg (2010) showed that the

original results were replicated across a large number of different data combinations, including three versions of the CRU surface data and two different satellite data sets. Consequently the claim that the findings were not repeatable across different data sets was shown to be untrue.

The IPCC was dismissive of the issue, based on an unsupported conjecture that there was a natural explanation for the correlation between warming and the spatial pattern of industrialization. McKittrick (2010a) tests this explanation and shows that inclusion of atmospheric oscillation patterns does not invalidate the correlations with socioeconomic trends.

The only other author who has critiqued the McKittrick-Michaels and deLaat-Maurellis results was Schmidt (2009). He argued that the surface temperature field exhibits spatial autocorrelation (SAC), an effect in which a trend in one grid cell has an effect that spills over to adjacent gridcells. This effect reduces the “effective degrees of freedom” in the sample and biases the test statistics towards concluding the data are contaminated. He also argued that use of the lower troposphere satellite series from Remote Sensing Systems, denoted RSS) rather than from the University of Alabama-Huntsville (denoted UAH), reduces the significance of the coefficients, indicating a lack of robustness of the conclusions. And he argued that the results were spurious on the basis of a comparison with results obtained by swapping the observed surface and tropospheric trends with model-generated data from NASA’s Goddard Institute of Space Studies (GISS) model E, denoted herein as GISS-E. These model-generated data are, by construction, uncontaminated by industrialization-induced surface changes. Schmidt’s hypothesis was that if the GISS-E data yield the same regression coefficients as the observational data in MM07, it would indicate that the seeming correlations between patterns of warming and patterns of industrialization were a fluke.

McKittrick and Nierenberg (2010) shows that the spatial autocorrelation argument of Schmidt is incorrect. Schmidt did not present any test statistics or re-estimations of the models. McKittrick and Nierenberg show that controlling for the effects of spatial autocorrelation did not undermine the test results, and in fact showed that the spatial structure of model predictions and observed data were quite distinct. In addition, the model-generated data did not yield a correlation pattern similar to that on observed data, instead McKittrick and Nierenberg showed that the model-generated pattern was opposite to that observed. Also, the use of RSS rather than UAH data did not undermine the conclusions: when a few outlier observations were removed the RSS results were even stronger than the original UAH-based ones.

The Benestad (2004) comment and the Schmidt (2009) papers were the ones relied upon by the University of East Anglia in its submission to the Muir Russell inquiry to defend the handling of this topic by Phil Jones in his capacity as IPCC Lead Author. The publication of the Schmidt (2009) paper is notable since it did not contain any statistical tests to support the assertions made in the abstract and conclusions. But even more problematic, the University of East Anglia cites Schmidt (2009) as independent support for Phil Jones’ CRU data, yet the climategate email archive shows Phil Jones was the reviewer of Schmidt’s paper for the journal in which it was published.<sup>8</sup>

Summarizing this topic:

- Two independent teams, publishing in four different climate journals, found evidence that socioeconomic correlations exist in climatic data, undermining the view that these patterns have been removed in the adjustment process.

---

<sup>8</sup> See [http://www.climate-gate.org/cru/documents/review\\_schmidt.doc](http://www.climate-gate.org/cru/documents/review_schmidt.doc).

- Unlike the Parker (2004, 2006) approach, the socioeconomic gridding approach directly measures activity that gives rise to urbanization and land surface modification.
- The papers that have been published arguing against these findings have themselves been rebutted.

## References

- Benestad R.E. (2004) Are temperature trends affected by economic activity? Comment on McKittrick & Michaels. (2004). *Climate Research* 27:171–173
- Brohan, P., Kennedy, J., Harris, I., Tett, S.F.B. and Jones, P.D., (2006): Uncertainty estimates in regional and global observed temperature changes: a new dataset from 1850. *J. Geophys. Res.* **111**, D12106, doi:10.1029/2005JD006548.
- Chagnon, S. A. (1999), A rare long record of deep soil temperature defines temporal temperature changes and an urban heat island, *Climatic Change* 42: 531—538.
- Christy, John R., David E. Parker, Simon J. Brown, Ian Macadam, Martin Stendel and William B. Norris (2001). “Differential Trends in Tropical Sea Surface and Atmospheric Temperatures since 1979.” *Geophysical Research Letters* 28(1), 183—186.
- Christy, J.R., W.B. Norris, K. Redmond, and K.P. Gallo (2006), “Methodology and results of calculating Central California surface temperature trends: Evidence of human-induced climate change?” *Journal of Climate*, 19, 548-563.
- De Laat, A.T.J., and A.N. Maurellis (2004). “Industrial CO2 Emissions as a Proxy for Anthropogenic Influence on Lower Tropospheric Temperature Trends.” *Geophysical Research Letters* Vol. 31, L05204, doi:10.1029/2003GL019024.
- De Laat, A.T.J., and A.N. Maurellis (2006). “Evidence for Influence of Anthropogenic Surface Processes on Lower Tropospheric and Surface Temperature Trends.” *International Journal of Climatology* 26:897—913.
- Folland, C.K. and D.E. Parker (1995) “Correction of Instrumental Biases in Historical Sea Surface Temperature Data.” *Quarterly Journal of the Royal Meteorological Society* (1995) **121**: 319-367.
- Hansen, J.E., and S. Lebedeff (1987) “Global trends of measured surface air temperature” *Journal of Geophysical Research* 92, 13345-13372, doi:10.1029/JD092iD11p13345.
- Hansen, J., R. Ruedy, J. Glascoe, and Mki. Sato (1999), “GISS analysis of surface temperature change,” *Journal of Geophysical Research*, 104, 30997-31022, doi:10.1029/1999JD900835.
- Hansen, J.E., R. Ruedy, Mki. Sato, M. Imhoff, W. Lawrence, D. Easterling, T. Peterson, and T. Karl (2001) “A closer look at United States and global surface temperature change” *Journal of Geophysical Research* 106, 23947-23963, doi:10.1029/2001JD000354.
- IPCC (2007) *Climate Change 2007: The Physical Science Basis. Contribution of Working Group I to the Fourth Assessment Report of the Intergovernmental Panel on Climate Change* S. Solomon, D. Qin, M. Manning, Z. Chen, M. Marquis, K.B. Averyt, M. Tignor and H.L. Miller (eds).. Cambridge University Press, Cambridge, United Kingdom and New York, NY, USA.
- Johnson, G.T., Oke, T.R., D.G. Steyn and I.D. Watson (1991) “Simulation of Surface Urban Heat Islands under ‘Ideal’ Conditions at Night Part 1: Theory and Tests against Field Data.” *Boundary-Layer Meteorology* 56 275—294.
- Jones, P.D. and A. Moberg (2003). “Hemispheric and Large-Scale Surface Air Temperature Variations: An Extensive Revision and an Update to 2001.” *Journal of Climate* 16 203—223.
- Jones, P.D., M. New, D. E. Parker, S. Martin, and I. G. Rigor, (1999) Surface air temperature and its changes over the past 150 years. *Reviews of Geophysics.*, **37**, 173–199.

- Jun, Mikyoung, Reto Knutti and Douglas W. Nychka (2008) Spatial analysis to quantify numerical model bias and dependence: How many climate models are there? *Journal of the American Statistical Association* 108 No. 483 934—947 DOI 10.1198/016214507000001265.
- Kalnay, E., Cai, M. (2004) “Reply to Trenberth and to Vose et al.” *Nature* 427 15 January 2004, 214.
- Kalnay, E., Cai, M. (2003) Impacts of urbanization and land-use change on climate, *Nature* 423, 528-531
- Kent, E., C. S.D. Woodruff and D.I. Berry (2007) “Metadata from WMO Publication No. 47 and an Assessment of Voluntary Observing Ship Observations in ICAODS. *Journal of Atmospheric and Ocean Technology* 24 DOI: 10.1175/JTECH1949.1.
- McKittrick, Ross R. (2010a) “Atmospheric Oscillations do not Explain the Temperature-Industrialization Correlation.” *Statistics, Politics and Policy* Vol 1, July 2010.
- McKittrick, Ross R. (2010b) Submission of Evidence to the Independent Climate Change Email Review. Available online at <http://www.cce-review.org>.
- McKittrick, Ross R. and Nicolas Nierenberg (2010). “Socioeconomic Signals in Climate Data.” *Journal of Economic and Social Measurement* Forthcoming.
- McKittrick, Ross R. and P.J. Michaels (2007) “Quantifying the influence of anthropogenic surface processes and inhomogeneities on gridded global climate data.” *Journal of Geophysical Research*, 112, D24S09, doi:10.1029/2007JD008465.
- McKittrick, Ross R. and Patrick J. Michaels (2004) “Are Temperature Trends Affected by Economic Activity? Reply to Benestad (2004b)” *Climate Research* 27(2) pp. 175-176.
- McKittrick, Ross R. and Patrick J. Michaels, (April 2004a) “A Test of Corrections for Extraneous Signals in Gridded Surface Temperature Data.” *Climate Research* 26(2) 159-173, April 2004.
- Oke, T.R. 1973. “City size and the urban heat island.” *Atmospheric Environment* 7: 769-779.
- Parker, D.E. (2006) “A Demonstration that Large-Scale Warming is not Urban.” *Journal of Climate* 19:2882—2895.
- Parker, D.E. (2004) “Climate: Large-Scale Warming is not Urban.” *Nature* 432, 290 (18 November 2004); doi:10.1038/432290a.
- Peterson, Thomas C., Russell Vose, Richard Schmoyer and Vyacheslav Razuvev (1998) “Global Historical Climatology Network (GHCN) Quality Control of Monthly Temperature Data.” *International Journal of Climatology* 18: 1169—1179.
- Peterson T.C. and R.S. Vose (1997) “An Overview of the Global Historical Climatology Network Temperature Database.” *Bulletin of the American Meteorological Society* 78:2837—2849.
- Pielke, RA Sr. and T. Matsui (2005) “Should Light Wind and Windy Nights have the Same Temperature Trends at Individual Levels Even if the Boundary Layer Averaged Heat Content Change is the Same?” *Geophysical Research Letters* (32) L21813, doi:10.1029/2005GL024407, 2005.
- Rayner, N. A., D. E. Parker, E. B. Horton, C. K. Folland, L. V. Alexander, D. P. Rowell, E. C. Kent, and A. Kaplan, (2003): Global analyses of sea surface temperature, sea ice, and night marine air temperature since the late nineteenth century. *Journal of Geophysical Research*, 108(D14), 4407, doi:10.1029/2002JD002670.
- Rayner, N.A., P. Brohan, D.E. Parker, C.K. Folland, J.J. Kennedy, M. Vanicek, T.J. Ansell and S.F.B. Tett (2006) “Improved Analyses of Changes and Uncertainties in Sea Surface Temperature Measured In Situ since the Mid-Nineteenth Century: The HadSST2 Dataset” *Journal of Climate* 19 446-469.
- Reynolds, Richard W., Thomas C. Peterson and Jay Lawrimore (2008). “Improvements to NOAA’s Historical Merged Land-Ocean Surface Temperature Analysis (1880—2006).” *Journal of Climate* DOI: 10.1175/2007JCLI2100.1.
- Schmidt, M. (2009) Spurious correlations between recent warming and indices of local economic activity. *International Journal of Climatology* DOI:10.1002/joc.1831.
- Simmons, A.J., P.D. Jones, V. da Costa Bechtold, A. C. M. Beljaars, P. W. Kallberg, S. Saarinen, S. M. Uppala, P. Viterbo, and N. Wedi (2004) “Comparison of trends and low-frequency variability in

- CRU, ERA-40, and NCEP//NCAR analyses of surface air temperature.” *Journal of Geophysical Research* 109 D24115, doi:10.1029/2004JD005306, 2004.
- Smith, Thomas M. and Richard W. Reynolds (2005) “A Global Merged Land–Air–Sea Surface Temperature Reconstruction Based on Historical Observations (1880–1997)” *Journal of Climate* 15 June 2005 2021—2036.
- Thompson, David W. J., John J. Kennedy, John M. Wallace and Phil D. Jones (2008). “A large discontinuity in the mid-twentieth century in observed global-mean surface temperature” *Nature* Vol 453, 29 May 2008 doi:10.1038/nature06982
- Trenberth, Kevin (2004) “Rural land-use change and climate” *Nature* 427 15 January 2004, 213.
- Vose, R. S., R. L. Schmoyer, P. M. Steurer, T. C. Peterson, R. Heim, T. R. Karl, and J. Eischeid, 1992: The Global Historical Climatology Network: Long-term monthly temperature, precipitation, sea level pressure, and station pressure data. ORNL/CDIAC-53, NDP-041, 325 pp.
- Vose, Russell, Thomas R.Karl, David R. Easterling, Claude N.Williams, Matthew J.Menne (2004) “Impact of land-use change on climate.” *Nature* 427 15 January 2004, 213-214.
- Woodruff, S.D., H.F. Diaz, E.C. Kent, R.W. Reynolds, and S.J. Worley (2008) “The evolving SST record from ICOADS.” In *Climate Variability and Extremes during the Past 100 Years* (S. Brönnimann, J. Luterbacher, T. Ewen, H.F. Diaz, R.S. Stolarski, and U. Neu, Eds.), Advances in Global Change Research, Vol. 33, Springer, 65-83 (DOI: 10.1007/978-1-4020-6766-2\_4).
- Woodruff, S.D., H. F. Diaz, S. J. Worley, R. W. Reynolds, and S. J. Lubker, (2005). “Early ship observational data and ICOADS.” *Climatic Change*, **73**, 169–194.
- Zhou, L., Dickinson, R, Tian, Y., Fang, J, Qingziang, L., Kaufman, R, Myneni, R., Tucker, C., (2004), Rapid Urbanization warming China’s climate faster than other areas, *Proceedings of the National Academy of Science*, June 29, 2004



## Author bio

Professor McKittrick holds a BA in economics from Queen's University, and an MA and Ph.D. in economics from the University of British Columbia. He is a Professor of Environmental Economics at the University of Guelph in Ontario, Canada. He is also a Senior Fellow of the Fraser Institute in Vancouver B.C., a member of the Academic Advisory Boards of the John Deutsch Institute in Kingston Ontario and the Global Warming Policy Foundation in London, UK, and an advisor to a Canadian foreign exchange consulting company.

His research areas include modeling the relationship between economic growth and pollution emissions; regulatory mechanism design; and various aspects of the science and policy of global warming. His economics research has appeared in such peer-reviewed journals as *The Journal of Environmental Economics and Management*, *Economic Modeling*, *The Canadian Journal of Economics*, *Empirical Economics*, *The Energy Journal*, and *Environmental and Resource Economics*. His physical science research has appeared in such peer-reviewed journals as *Journal of Geophysical Research*, *Geophysical Research Letters*, *Climate Research*, *The Journal of Non-Equilibrium Thermodynamics* and *Proceedings of the National Academy of Sciences*. He is the author of the advanced textbook *Economic Analysis of Environmental Policy*, to be published by the University of Toronto Press in fall 2010. In 2002 he and Christopher Essex of the University of Western Ontario published the book *Taken By Storm: The Troubled Science, Policy and Politics of Global Warming* which was awarded the \$10,000 Donner Prize for Best Book on Canadian Public Policy.

Professor McKittrick is widely-cited in Canada and around the world as an expert on global warming and environmental policy issues. He has been interviewed by numerous media outlets including Time, The New York Times, The Wall Street Journal, The National Post, The Globe and Mail, the CBC, BBC, ITV, Fox News, Bloomberg, Global TV, CTV, and others. His commentaries have appeared in many newspapers and magazines, including Newsweek and the Financial Post. His research has been discussed in such places as Nature, Science, The Economist, *Natuurwetenschap&Techniek*, The National Post, The Globe and Mail and in a front page article in the The Wall Street Journal (Feb 14 2005).

Professor McKittrick has made invited academic presentations in Canada, the US and Europe, and he has testified before the US Congress and the Canadian Parliamentary Finance and Environment Committees. In 2006 he was one of 12 experts from around the world asked to brief a panel of the US National Academy of Sciences on paleoclimate reconstruction methodology.

Professor McKittrick thanks Chad Herman (<http://treesfortheforest.wordpress.com/>) for computational support, including analysis of GHCN archives and preparation of graphs.

## Appendix: Computational details

For the computations using GHCN and related data sets the following source files were used:  
 GISTEMP: SBBX.Tsurf250, SBBX.Tsurf1200 <ftp://data.giss.nasa.gov/pub/gistemp/download/>  
 CRU: CRUTEM3.nc <http://hadobs.metoffice.com/crutem3/data/CRUTEM3.nc>  
 GHCN: v2.mean, v2.mean\_adj <ftp://ftp.ncdc.noaa.gov/pub/data/ghcn/v2>  
 NOAA: grid\_1880\_2010.dat <ftp://ftp.ncdc.noaa.gov/pub/data/ghcn/v2/grid/>

### MAIN PROGRAM

```
PROGRAM ghcn_main
! Purpose:
! This program will load the GHCN mean, min or maximum monthly
! temperature data, combine duplicate station records and
! grid the results.
!
! Record of revisions
! Date          Programmer      Description of change
! -----
! May 10, 2010   Chad Herman    Original code

USE ghcn_import, ONLY: get_parameters, get_stations, load_station_data, get_station_indices
USE ghcn_stat_comb, ONLY: consolidate_duplicates
USE ghcn_grid_func, ONLY: gen_grid, gen_grid_index, grid_data
USE ghcn_anomaly
USE ghcn_write, ONLY: write_netCDF

IMPLICIT NONE

CALL get_parameters
CALL get_stations
CALL load_station_data
CALL get_station_indices
CALL consolidate_duplicates
CALL gen_grid
CALL gen_grid_index
CALL grid_data
CALL get_anomaly
CALL write_netCDF

END PROGRAM ghcn_main
```

### GHCN\_IMPORT

```
MODULE ghcn_import

CONTAINS

!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!

SUBROUTINE get_nlines(file, nlines)
! Purpose:
! This subroutine counts the number of lines in a given file.
!
! Record of revisions
! Date          Programmer      Description of change
! -----
! May 10, 2010   Chad Herman    Original code
! July 10, 2010  Chad Herman    Replaced USE constants with
! USE numeric_kinds.

USE numeric_kinds

IMPLICIT NONE

CHARACTER(LEN=*) INTENT(IN) :: file
INTEGER(KIND=i4b), INTENT(OUT) :: nlines

INTEGER :: ioerr

! Open the file. If it can't be opened then terminate the program.
OPEN(UNIT = 1, FILE = file, STATUS = 'OLD', IOSTAT = ioerr)
IF(ioerr /= 0) THEN
  WRITE(*, fmt = '(a) ') 'Unable to open' // file
  STOP
ENDIF

! Initialize the nlines variable to zero and begin reading line by line
! accumulating in nlines until the end of the file is reached.
nlines = 0
DO WHILE(ioerr == 0)
  READ(UNIT = 1, FMT = *, IOSTAT = ioerr)
  IF(ioerr == 0) THEN
    nlines = nlines + 1
  ENDIF
ENDDO

! Close the file.
CLOSE(UNIT = 1)
```

```

END SUBROUTINE get_nlines

!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!

SUBROUTINE get_parameters
! Purpose:
! This subroutine loads the parameters file that contains the input/output
! file names, start and end year for calculating the climatology and
! minimum number of months required to calculate the climatology.
!
! Record of revisions
! Date           Programmer           Description of change
! -----
! May 11, 2010   Chad Herman           Original code
! May 24, 2010   Chad Herman           Added read statement to get the
! preferred start and end years.
! July 10, 2010   Chad Herman           Replaced USE constants with
! USE numeric_kinds. Replaced KIND=lg with KIND=i4b.

USE numeric_kinds
USE ghc_n_parameters, ONLY: outfile, year0_b, yearf_b, min_n, parameter_file,&
    stat_list_file, stat_data_file, dlon, dlat,&
    start_year, end_year

IMPLICIT NONE

INTEGER(KIND=i4b) :: ioerr
LOGICAL(KIND=i4b) :: exists, cont
CHARACTER(LEN=1) :: response

OPEN(UNIT = 1, FILE = parameter_file, STATUS = 'OLD', IOSTAT = ioerr)
IF(ioerr /= 0)THEN
    WRITE(UNIT = *, FMT = '(A)')'Unable to open the parameter file ' // parameter_file
    STOP
ENDIF

READ(UNIT = 1, FMT = *, IOSTAT = ioerr)stat_list_file
IF(ioerr /= 0)THEN
    WRITE(UNIT = *, FMT = '(A)')'Unable to read the parameter file ' // parameter_file // &
    ' to read the stat_list_file variable.'
    STOP
ENDIF

READ(UNIT = 1, FMT = *, IOSTAT = ioerr)stat_data_file
IF(ioerr /= 0)THEN
    WRITE(UNIT = *, FMT = '(A)')'Unable to read the parameter file ' // parameter_file // &
    ' to read the stat_data_file variable.'
    STOP
ENDIF

READ(UNIT = 1, FMT = *, IOSTAT = ioerr)outfile
IF(ioerr /= 0)THEN
    WRITE(UNIT = *, FMT = '(A)')'Unable to read the parameter file ' // parameter_file // &
    ' to read the outfile variable.'
    STOP
ENDIF

READ(UNIT = 1, FMT = *, IOSTAT = ioerr)dlon
IF(ioerr /= 0)THEN
    WRITE(UNIT = *, FMT = '(A)')'Unable to read the parameter file ' // parameter_file // &
    ' to read the dlon variable.'
    STOP
ENDIF

READ(UNIT = 1, FMT = *, IOSTAT = ioerr)dlat
IF(ioerr /= 0)THEN
    WRITE(UNIT = *, FMT = '(A)')'Unable to read the parameter file ' // parameter_file // &
    ' to read the dlat variable.'
    STOP
ENDIF

READ(UNIT = 1, FMT = *, IOSTAT = ioerr)year0_b
IF(ioerr /= 0)THEN
    WRITE(UNIT = *, FMT = '(A)')'Unable to read the parameter file ' // parameter_file // &
    ' to read the year0_b variable.'
    STOP
ENDIF

READ(UNIT = 1, FMT = *, IOSTAT = ioerr)yearf_b
IF(ioerr /= 0)THEN
    WRITE(UNIT = *, FMT = '(A)')'Unable to read the parameter file ' // parameter_file // &
    ' to read the yearf_b variable.'
    STOP
ENDIF

READ(UNIT = 1, FMT = *, IOSTAT = ioerr)min_n
IF(ioerr /= 0)THEN
    WRITE(UNIT = *, FMT = '(A)')'Unable to read the parameter file ' // parameter_file // &
    ' to read the min_n variable.'
    STOP
ENDIF

READ(UNIT = 1, FMT = *, IOSTAT = ioerr)start_year
IF(ioerr /= 0)THEN
    WRITE(UNIT = *, FMT = '(A)')'Unable to read the parameter file ' // parameter_file // &
    ' to read the start_year variable.'
    STOP
ENDIF

READ(UNIT = 1, FMT = *, IOSTAT = ioerr)end_year
IF(ioerr /= 0)THEN

```

```

WRITE(UNIT = *, FMT = '(A)')'Unable to read the parameter file ' // parameter_file // &
' to read the end_year variable.'
STOP
ENDIF
ENDIF

CLOSE(UNIT = 1)

! Check if the output file already exists. If it does, ask the user if it is OK to overwrite.
INQUIRE(FILE = outfile, EXIST = exists)

cont = .TRUE.
! If it does then ask the user if they want to overwrite the file.
IF(exists)THEN

DO WHILE(cont)

WRITE(UNIT = *, FMT = '(A)')'The file ' // TRIM(outfile) // ' already exists.'
WRITE(UNIT = *, FMT = '(A)', ADVANCE = 'NO')'Do you want to overwrite? (Y/N) '
READ(UNIT = *, FMT = '(A1)')response

IF(response == 'N' .OR. response == 'n')THEN
WRITE(UNIT = *, FMT = '(A)')'Program terminated.'
STOP
ELSEIF(response == 'Y' .OR. response == 'y')THEN
cont = .FALSE.
ELSE
WRITE(UNIT = *, FMT = '(A)')'Unrecognized response. Try again.'
WRITE(UNIT = *, FMT = '(A)')
ENDIF

ENDDO

ENDIF

END SUBROUTINE get_parameters

!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!

SUBROUTINE get_stations
! Purpose:
! This subroutine loads the GHCN temperature inventory file into memory.
!
! Record of revisions
! Date          Programmer      Description of change
! -----
! May 10, 2010  Chad Herman      Original code
! May 21, 2010  Chad Herman      Added variable station%brightness
! July 10, 2010 Chad Herman      Replaced USE constants with
! USE numeric_kinds.

USE numeric_kinds
USE ghcn_data, ONLY : stations
USE ghcn_parameters, ONLY : stat_list_file, stat_list_fmt

IMPLICIT NONE

INTEGER(KIND=i4b) :: nlines, err, ioerr, i

WRITE(UNIT = *, FMT = '(A)', ADVANCE = 'NO')'Loading inventory data ... '

! Get the number of lines in the inventory file.
CALL get_nlines(stat_list_file, nlines)

! Now allocate memory for the meta type variable.
ALLOCATE(stations(nlines), STAT = err)
IF(err /= 0)THEN
WRITE(*, FMT = '(A)')'Unable to allocate memory for the station inventory.'
STOP
ENDIF
ENDIF

! Open the inventory file.
OPEN(UNIT = 1, FILE = stat_list_file, STATUS = 'OLD', IOSTAT = ioerr)
IF(ioerr /= 0)THEN
WRITE(UNIT = *, FMT = '(A)')'Unable to open the station inventory file ' // stat_list_file
STOP
ENDIF

! Begin reading in the data.
DO i = 1, nlines

READ(UNIT = 1, FMT = stat_list_fmt, IOSTAT = ioerr)&
stations(i)%cc,&
stations(i)%wmo,&
stations(i)%mod,&
stations(i)%name,&
stations(i)%lat,&
stations(i)%lon,&
stations(i)%elev,&
stations(i)%elev_int,&
stations(i)%pop_char,&
stations(i)%pop,&
stations(i)%topo,&
stations(i)%veg,&
stations(i)%loc,&
stations(i)%loc_coast,&
stations(i)%airport,&
stations(i)%airtowndis,&
stations(i)%grveg,&
stations(i)%brightness

IF(ioerr /= 0)THEN

```

```

        WRITE(UNIT = *, FMT = '(A)')'Unable to read the inventory file ' // stat_list_file
        STOP
    ENDIF

ENDDO

! Close the file.
CLOSE(UNIT = 1)

WRITE(UNIT = *, FMT = '(A)')'Done.'

END SUBROUTINE get_stations

!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!

SUBROUTINE load_station_data
! Purpose:
! This subroutine loads the GHCN temperature data into memory.
!
! Record of revisions
! Date           Programmer       Description of change
! -----
! May 10, 2010   Chad Herman       Original code
! May 24, 2010   Chad Herman       Code modified to allow loading only
! a subset of the data by year.
! July 10, 2010  Chad Herman       Replaced USE constants with
! USE numeric_kinds. Replaced KIND=lg with KIND=i4b.

USE numeric_kinds
USE ghcn_data, ONLY : cc, wmo, mod, dup, year, temperature
USE ghcn_parameters, ONLY : stat_data_file, stat_data_fmt, missing_int,&
                           ghcn_missing, stat_data_fmt_year, start_year,&
                           end_year

IMPLICIT NONE

INTEGER(KIND=i4b) :: nlines, err, ioerr, i, k
INTEGER(KIND=i2b), ALLOCATABLE, DIMENSION(:) :: years_present
INTEGER(KIND=i4b) :: n_years_present, ntime
LOGICAL(KIND=i4b) :: subset

WRITE(UNIT = *, FMT = '(A)', ADVANCE = 'no')'Loading temperature data ... '

! Get the number of lines in the data file.
CALL get_nlines(stat_data_file, nlines)

! Open the data file.
OPEN(UNIT = 1, FILE = stat_data_file, STATUS = 'OLD', IOSTAT = ioerr)
IF(ioerr /= 0)THEN
    WRITE(*,fmt='(a)')'Unable to open ' // stat_data_file
    STOP
ENDIF

! If both start_year and end_year are defined then read through the data file and
! collect the years.
IF(start_year /= -1 .AND. end_year /= -1)THEN

    subset = .TRUE.
    ALLOCATE(years_present(nlines), STAT = err)

    DO i = 1, nlines
        READ(UNIT = 1, FMT = stat_data_fmt_year, IOSTAT = ioerr)years_present(i)
        IF(ioerr /= 0)THEN
            WRITE(*,fmt='(a)')'Unable to read ' // stat_data_file
            STOP
        ENDIF
    ENDDO

    REWIND(UNIT = 1)

! Get the number of years that are within the requested range. If there are none, then
! terminate.
    n_years_present = COUNT(years_present >= start_year .AND. years_present <= end_year)
    IF(n_years_present == 0)THEN
        WRITE(UNIT = *, FMT = '(A)')'There are no years of data on the subset requested.'
        STOP
    ENDIF

    ntime = n_years_present

    DEALLOCATE(years_present)

ELSE

    subset = .FALSE.
    ntime = nlines

ENDIF

! Now allocate memory for the data.
ALLOCATE(cc(ntime),&
        wmo(ntime),&
        mod(ntime),&
        dup(ntime),&
        year(ntime),&
        temperature(ntime,12),&
        STAT = err)
IF(err /= 0)THEN
    WRITE(*,fmt='(a)')'Unable to allocate memory for stat_data'
    STOP
ENDIF

```

```

k = 1
! Read in the data.
DO i = 1, nlines

  IF(subset .EQV. .FALSE.)THEN
    k = i
  ENDIF

  READ(UNIT = 1, FMT = stat_data_fmt, IOSTAT = ioerr)&
  cc(k),&
  wmo(k),&
  mod(k),&
  dup(k),&
  year(k),&
  temperature(k,:)

  IF(ioerr /= 0)THEN
    WRITE(*,fmt='(a)')'Unable to read ' // stat_data_file
    STOP
  ENDIF

  IF(subset .AND. year(k) >= start_year .AND. year(k) <= end_year)THEN
    k = k + 1
  ENDIF

ENDDO

CLOSE(UNIT = 1)

WHERE(temperature == ghcn_missing) temperature = missing_int

WRITE(UNIT = *,FMT = '(A)')'Done.'

END SUBROUTINE load_station_data

!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!

SUBROUTINE get_station_indices
! Purpose:
! This subroutine generates lower and upper indices that correspond to
! each wmo/mod pair to facilitate the extraction of data from the
! 'year' and 'temperature' arrays.
!
! Record of revisions
! Date           Programmer           Description of change
! -----
! May 10, 2010   Chad Herman           Original code
! May 21, 2010   Chad Herman           Added an initialization of
! the index_0 and index_f variables to a missing value so if the
! station is not found, then the other subroutines will know not
! to use the station.
! July 10, 2010   Chad Herman           Replaced USE constants with
! USE numeric_kinds. Replaced KIND=lg with KIND=i4b.

USE numeric_kinds
USE ghcn_data, ONLY : stations, wmo, mod

IMPLICIT NONE

INTEGER(KIND=i4b) :: i, j
LOGICAL(KIND=i4b) :: first, cont

WRITE(UNIT = *, FMT = '(A)', ADVANCE = 'no')'Generating station lookup indices ... '

j = 1
DO i = 1, SIZE(stations)

  first = .TRUE.
  cont = .TRUE.

  stations(i)%index_0 = - 1
  stations(i)%index_f = - 1

  DO WHILE(cont)

    IF(stations(i)%wmo == wmo(j) .AND. &
      & stations(i)%mod == mod(j))THEN

      IF(first)THEN
        stations(i)%index_0 = j
        first = .FALSE.
      ELSE
        stations(i)%index_f = j
      ENDIF

      j = j + 1

    ELSE

      cont = .FALSE.

    ENDIF

  ENDDO

  IF(stations(i)%index_f == -1)THEN
    stations(i)%index_f = stations(i)%index_0
  ENDIF

ENDDO

```

```
WRITE(UNIT = *, FMT = '(A)') 'Done.'
```

```
END SUBROUTINE get_station_indices
```

```
!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!
```

```
END MODULE ghcn_import
```

#### STAT\_COMB

```
MODULE ghcn_stat_comb
```

```
CONTAINS
```

```
!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!
```

```
SUBROUTINE consolidate_duplicates
```

```
! Purpose:
```

```
! This subroutine will consolidate the multiple records that often appear  
! from the same station.
```

```
!
```

```
! Record of revisions
```

```
! Date           Programmer           Description of change
```

```
! -----
```

```
! May 10, 2010   Chad Herman           Original code
```

```
! July 10, 2010  Chad Herman           Replaced USE constants with
```

```
! USE numeric_kinds. Changed KIND=r8b to KIND=r8b. Replaced KIND=lg with
```

```
! KIND=i4b. Replaced _r8b with _r8b.
```

```
USE numeric_kinds
```

```
USE ghcn_data, ONLY: stations, dup, year, temperature
```

```
USE ghcn_parameters, ONLY: missing_int
```

```
IMPLICIT NONE
```

```
INTEGER(KIND=i4b) :: nmonths
```

```
LOGICAL(KIND=i4b), DIMENSION(0:9) :: duplicate
```

```
INTEGER(KIND=i4b), DIMENSION(0:9) :: duplicate_index
```

```
INTEGER(KIND=i4b) :: ndup
```

```
REAL(KIND=r8b), ALLOCATABLE, DIMENSION(:, :, :) :: temp_mat
```

```
REAL(KIND=r8b), ALLOCATABLE, DIMENSION(:, :, :) :: temp_col
```

```
INTEGER(KIND=i4b), DIMENSION(1:12) :: station_data_year
```

```
INTEGER(KIND=i4b), ALLOCATABLE, DIMENSION(:) :: indices
```

```
INTEGER(KIND=i4b) :: i, j, k
```

```
INTEGER(KIND=i4b) :: err
```

```
WRITE(UNIT = *, FMT = '(A)', ADVANCE = 'NO') 'Consolidating duplicate stations ... '
```

```
DO i = 1, SIZE(stations)
```

```
IF(stations(i)%index_0 /= -1) THEN
```

```
! Initialize the duplicate array
```

```
duplicate = .FALSE.
```

```
! Fill in the duplicate spots that correspond to existing duplicates for
```

```
! stations(i).
```

```
DO j = 0, 9
```

```
IF(ANY(dup(stations(i)%index_0:stations(i)%index_f) == j)) THEN
```

```
duplicate(j) = .TRUE.
```

```
ENDIF
```

```
ENDDO
```

```
! Count how many duplicate there are and create an index that tells what column  
! each duplicate should be stored. This is done because the list of duplicates  
! may be discontinuous. For example, the list could be: 0, 1, 4, 5. These data  
! should be stored not in columns 0, 1, 4 and 5 because that would leave columns  
! 2 and 3 empty. They should be stored in columns 0, 1, 2 and 3.
```

```
ndup = 0
```

```
DO j = 0, 9
```

```
IF(duplicate(j)) THEN
```

```
duplicate_index(j) = ndup
```

```
ndup = ndup + 1
```

```
ENDIF
```

```
ENDDO
```

```
! Get the first and last year across all duplicates.
```

```
stations(i)%year_0 = MINVAL(year(stations(i)%index_0:stations(i)%index_f))
```

```
stations(i)%year_f = MAXVAL(year(stations(i)%index_0:stations(i)%index_f))
```

```
! Calculate the number of months covered by all the duplicates.
```

```
nmonths = 12*(stations(i)%year_f - stations(i)%year_0 + 1)
```

```
! Allocate memory for the arrays to contain the temperature data.
```

```
ALLOCATE(stations(i)%temperature(nmonths), stat = err)
```

```
IF(err /= 0) THEN
```

```
WRITE(*, FMT = '(A)') 'Unable to allocate memory for stations(i)%temperature(nmonths).'
```

```
STOP
```

```
ENDIF
```

```
ALLOCATE(temp_mat(stations(i)%year_0:stations(i)%year_f, 1:12, 0:(ndup-1)), stat = err)
```

```
IF(err /= 0) THEN
```

```
WRITE(*, FMT = '(A)') 'Unable to allocate memory for temp_mat.'
```

```
STOP
```

```
ENDIF
```

```
ALLOCATE(temp_col(nmonths, 0:(ndup-1)), stat = err)
```

```
IF(err /= 0) THEN
```

```
WRITE(*, FMT = '(A)') 'Unable to allocate memory for temp_col.'
```

```
STOP
```

```

ENDIF
! Initialize the temp_mat array with original GHCN missing values.
temp_mat = missing_int

! Load the data for stations(i) into temp_mat. Loading it this way is easy because the
! year acts as the index.

DO j = stations(i)%index_0, stations(i)%index_f
  station_data_year = temperature(j,:)
  DO k = 0, 9
    IF (dup(j) == k) THEN
      temp_mat(year(j), 1:12, duplicate_index(k)) = station_data_year
      EXIT
    ENDIF
  ENDDO
ENDDO

! Scale the data by 1/10 to convert to degree Celsius.
temp_mat = temp_mat/10.0_r8b

! Copy temp_mat into temp_col
DO j = 0, ndup - 1
  temp_col(:,j) = RESHAPE(TRANPOSE(temp_mat(:,j)), (/nmonths/))
ENDDO

! If there's only one existing duplicate, then just copy the data into stations(i).
! Otherwise, sent the data to combine_stations. First combine only the January data,
! then the February data second, etc.
IF (ndup == 1) THEN

  stations(i)%temperature = temp_col(:,duplicate_index(0))

ELSE

  ALLOCATE(indices(nmonths/12), STAT = err)
  IF (err /= 0) THEN
    WRITE(*, FMT = '(A)') 'Unable to allocate memory for indices.'
    STOP
  ENDIF

  DO j = 1, 12
    indices = seq(j, nmonths, 12)
    stations(i)%temperature(indices) = combine_stations(temp_col(indices,:))
  ENDDO

  DEALLOCATE(indices)

ENDIF

DEALLOCATE(temp_col)
DEALLOCATE(temp_mat)

ENDIF
ENDDO

WRITE(UNIT = *, FMT = '(A)') 'Done.'
END SUBROUTINE consolidate_duplicates

!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!

FUNCTION seq(x0, xf, x_step) RESULT(x)
! Purpose:
! This function generates a sequence of numbers starting with x0 and
! proceeding in steps of x_step.
!
! Record of revisions
! Date           Programmer      Description of change
! -----
! May 10, 2010   Chad Herman    Original code
! July 10, 2010 Chad Herman    Replaced USE constants with
! USE numeric_kinds.

USE numeric_kinds

IMPLICIT NONE

INTEGER(KIND=i4b), INTENT(IN) :: x0, xf, x_step
INTEGER(KIND=i4b), DIMENSION( (xf - x0)/x_step + 1 ) :: x

INTEGER(KIND=i4b) :: i

x(1) = x0
DO i = 2, SIZE(x)
  x(i) = x(i-1) + x_step
ENDDO

END FUNCTION seq

!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!

INTEGER FUNCTION common_obs(x1, x2)
! Purpose:
! This function calculates the number of common observations between
! two temperature time series.
!
! Record of revisions
! Date           Programmer      Description of change
! -----
! May 10, 2010   Chad Herman    Original code

```



```

! July 10, 2010      Chad Herman      Added USE numeric_kinds. Replaced
! KIND=r8b with KIND=r8b. Replaced eps_d with eps_r8b.

USE constants
USE numeric_kinds
USE ghcn_parameters, ONLY: missing_dec

IMPLICIT NONE

REAL(KIND=r8b), DIMENSION(:), INTENT(in) :: x1
REAL(KIND=r8b), DIMENSION(:), INTENT(in) :: x2

INTEGER(KIND=i4b) :: i

! Check to make sure the sizes of x1 and x2 match.
IF(SIZE(x1,1) /= SIZE(x2,1))THEN
  WRITE(*,fmt='(a)')'The two stations passed to common_obs are different sizes.'
  STOP
ENDIF

common_obs = 0

! Loop over all time steps and compare both stations to see if their data
! is non-missing.
DO i = 1, SIZE(x1)
  IF( ABS(x1(i) - missing_dec) > eps_r8b .AND. &
    & ABS(x2(i) - missing_dec) > eps_r8b )THEN
    common_obs = common_obs + 1
  ENDIF
ENDDO

END FUNCTION common_obs

!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!

FUNCTION get_N(stat_in) RESULT(N)
! Purpose:
! This function generates the design matrix for the optimization problem
! involving calculate offsets for combining multiple stations.
!
! Record of revisions
! Date           Programmer      Description of change
! -----
! May 10, 2010   Chad Herman      Original code
! July 10, 2010   Chad Herman      Replaces USE constants with
! USE numeric_kinds. Replaced KIND=r8b with KIND=r8b.

USE numeric_kinds

IMPLICIT NONE

REAL(KIND=r8b), DIMENSION(:, :), INTENT(in) :: stat_in
REAL(KIND=r8b), DIMENSION(SIZE(stat_in,2), SIZE(stat_in,2)) :: N

INTEGER(KIND=i4b), DIMENSION(SIZE(stat_in,2), SIZE(stat_in,2)) :: N_temp
INTEGER(KIND=i4b) :: nstat
INTEGER(KIND=i4b) :: i, j, k

nstat = SIZE(stat_in,2)

! Initialize the design matrix to zero.
N_temp = 0

! It's best to read the documentation of how the design matrix is constructed
! or just look at the code and you'll figure it out yourself.
DO i = 1, nstat
  DO j = 1, nstat

    IF(i /= j)THEN
      N_temp(i,j) = -common_obs(stat_in(:,i),stat_in(:,j))
    ELSE

      DO k = 1, nstat
        IF(k /= i)THEN
          N_temp(i,j) = N_temp(i,j) + common_obs(stat_in(:,i),stat_in(:,k))
        ENDIF
      ENDDO

    ENDIF

  ENDDO

ENDDO

! Now convert the data type to a double precision real.
N = 1.0_r8b*N_temp

END FUNCTION get_N

!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!

FUNCTION get_S(stat_in) RESULT(S)
! Purpose:
! This function calculates the response vector in relation to the design
! matrix N. See documentation elsewhere for more details.
!
! Record of revisions
! Date           Programmer      Description of change
! -----
! May 10, 2010   Chad Herman      Original code
! July 10, 2010   Chad Herman      Added USE numeric_kinds.
! Replaced KIND=r8b with KIND=r8b. Replaced eps_d with eps_r8b.

```

```

USE constants
USE numeric_kinds
USE ghcn_parameters

IMPLICIT NONE

REAL(KIND=r8b), DIMENSION(:,.), INTENT(in) :: stat_in
REAL(KIND=r8b), DIMENSION(SIZE(stat_in,2)) :: S

INTEGER(KIND=i4b) :: i, j, k
INTEGER(KIND=i4b) :: nstat, nobs

nstat = SIZE(stat_in,2)
nobs = SIZE(stat_in,1)

! Initialize the S vector to zero.
S = 0.0

! Calculate the sum of differences between each station combination.
DO i = 1, nstat
  DO j = 1, nstat
    DO k = 1, nobs
      IF ( ABS(stat_in(k,i) - missing_dec) > eps_r8b .AND. &
          & ABS(stat_in(k,j) - missing_dec) > eps_r8b ) THEN

        S(i) = S(i) + (stat_in(k,i) - stat_in(k,j))

      ENDIF
    ENDDO
  ENDDO
ENDDO

END FUNCTION get_S

!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!

FUNCTION diag(x) RESULT(x_d)
! Purpose:
! This function takes vector x and replicates it on the diagonal of a
! matrix whose sides are the length of x.
!
! Record of revisions
! Date           Programmer           Description of change
! -----
! May 10, 2010   Chad Herman       Original code
! July 10, 2010 Chad Herman       Replace USE constants with
! USE numeric_kinds. Replaced KIND=r8b with KIND=r8b. Replaced _r8b with
! _r8b.

USE constants

REAL(KIND=r8b), DIMENSION(:), INTENT(in) :: x
REAL(KIND=r8b), DIMENSION(size(x),size(x)) :: x_d

INTEGER(KIND=i4b) :: i, j

DO i = 1, size(x)
  DO j = 1, size(x)
    IF(i == j) THEN
      x_d(i,j) = x(i)
    ELSE
      x_d(i,j) = 0.0_r8b
    ENDIF
  ENDDO
ENDDO

END FUNCTION diag

!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!

SUBROUTINE svd(A, u, v, s)
! Purpose:
! This subroutine computes performs SVD on matrix A to facilitate the
! inversion of the design matrix N.
!
! Record of revisions
! Date           Programmer           Description of change
! -----
! May 10, 2010   Chad Herman       Original code
! July 10, 2010 Chad Herman       Replace USE constants with
! USE numeric_kinds. Replaced KIND=r8b with KIND=r8b.

USE LA_PRECISION, ONLY: WP => DP
USE F95_LAPACK, ONLY: LA_GESDD
USE numeric_kinds

IMPLICIT NONE

REAL(KIND=r8b), DIMENSION(:,.), INTENT(INOUT) :: A
REAL(KIND=r8b), DIMENSION(:,.), INTENT(OUT) :: u, v
REAL(KIND=r8b), DIMENSION(SIZE(A,1),SIZE(A,1)) :: vt
REAL(KIND=r8b), DIMENSION(:), INTENT(OUT) :: s
INTEGER(KIND=i4b) :: info

CALL la_gesdd(A, s, u = u, vt = vt, info = info)

IF(info > 0) THEN
  WRITE(*, FMT = '(a,i3,a)') 'Argument ', info, ' had an illegal value.'

```

```

      STOP
    ELSE IF(info < 0)THEN
      WRITE(*, FMT = '(a)')'The algorithm did not converge.'
      STOP
    ENDIF

    v = TRANSPOSE(vt)

  END SUBROUTINE svd

!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!

FUNCTION combine_stations(x) RESULT(x_comb)
! Purpose:
! This function accepts a number of stations, computes seasonal offsets
! to adjust each series to bring them into alignment for averaging into
! one series.
!
! Record of revisions
! Date           Programmer       Description of change
! -----
! May 10, 2010   Chad Herman       Original code
! July 10, 2010 Chad Herman       Added USE numeric_kinds.
! Replaced KIND=r8b with KIND=r8b. Replaced eps_d with eps_r8b. Replaced
! KIND=lg with KIND=i4b.

  USE constants
  USE numeric_kinds
  USE ghcn_parameters, ONLY: missing_dec

  REAL(KIND=r8b), DIMENSION(:,:), INTENT(IN) :: x
  REAL(KIND=r8b), DIMENSION(SIZE(x,1)) :: x_comb

  REAL(KIND=r8b), DIMENSION(SIZE(x,2),SIZE(x,2)) :: N, N_inv, u, v
  REAL(KIND=r8b), DIMENSION(SIZE(x,2)) :: S, sing
  REAL(KIND=r8b), DIMENSION(SIZE(x,2)) :: offsets

  INTEGER(KIND=i4b) :: i, ntime, nstat, ncount
  LOGICAL(KIND=i4b), DIMENSION(SIZE(x,2)) :: real_mask

  ntime = SIZE(x,1)
  nstat = SIZE(x,2)
  N = get_N(x)
  S = get_S(x)

! Use singular value decomposition to begin to invert the matrix N.
CALL svd(N, u, v, sing)

! Find entries that look like zero and set them to zero.
! All other entries become reciprocals of themselves.
WHERE(sing < 1000.0_r8b*eps_r8b)
  sing = 0.0_r8b
ELSEWHERE
  sing = 1/sing
ENDWHERE

! Now get the inverted matrix N_inv.
N_inv = MATMUL(MATMUL(v, diag(sing)), TRANSPOSE(v))

! Multiple N_inv with S to get the offsets.
offsets = SUM(N_inv*RESHAPE(S, (/ nstat, nstat/), PAD = S, ORDER = (/2, 1/)), DIM = 2)

! Now combine the stations with the given offsets.
DO i = 1, ntime

  real_mask = ABS(x(i,:) - missing_dec) > eps_r8b
  x_comb(i) = SUM(x(i,:) - offsets, MASK = real_mask)
  ncount = COUNT(real_mask)

  IF(ncount > 0)THEN
    x_comb(i) = x_comb(i)/ncount
  ELSE
    x_comb(i) = missing_dec
  ENDIF

ENDDO

END FUNCTION combine_stations

!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!

END MODULE ghcn_stat_comb

```

**GRID\_FUNC**

```

MODULE ghcn_grid_func

CONTAINS

!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!

SUBROUTINE gen_grid
! Purpose:
! This subroutine generates a grid based on the values of dlon/dlat. It
! calculates the latitude/longitude values as well as their bounds and
! an array of normalize surface area bound by the gridboxes.
!
! Record of revisions
! Date           Programmer       Description of change
! -----
! May 10, 2010   Chad Herman       Original code

```

```

USE ghcn_parameters, ONLY: dlon, dlat
USE ghcn_data, ONLY: grid

IMPLICIT NONE

INTEGER :: err
INTEGER :: i

WRITE(UNIT = *, FMT = '(A)', ADVANCE = 'NO')'Generating grid ... '

! Copy the incoming dlon/dlat into the grid_parameters type variable.
grid%dlon = dlon
grid%dlat = dlat

! Calculate the number of grid points.
grid%nlon = nint(360.0/grid%dlon)
grid%nlat = nint(180.0/grid%dlat)

! Allocate memory for all the allocatable variables present in the
! grid_parameters type variable.
ALLOCATE(grid%lon(grid%nlon), STAT = err)
IF(err /= 0)THEN
  WRITE(*,fmt='(a)')'Unable to allocate memory for grid%lon.'
  STOP
ENDIF

ALLOCATE(grid%lat(grid%nlat), STAT = err)
IF(err /= 0)THEN
  WRITE(*,fmt='(a)')'Unable to allocate memory for grid%lat.'
  STOP
ENDIF

ALLOCATE(grid%lon_bounds(2,grid%nlon), STAT = err)
IF(err /= 0)THEN
  WRITE(*,fmt='(a)')'Unable to allocate memory for grid%lon_bounds.'
  STOP
ENDIF

ALLOCATE(grid%lat_bounds(2,grid%nlat), STAT = err)
IF(err /= 0)THEN
  WRITE(*,fmt='(a)')'Unable to allocate memory for grid%lat_bounds.'
  STOP
ENDIF

ALLOCATE(grid%area(grid%nlon,grid%nlat), STAT = err)
IF(err /= 0)THEN
  WRITE(*,fmt='(a)')'Unable to allocate memory for grid%area.'
  STOP
ENDIF

! Now generate the grid's lat/lon values as well as their bounds.
grid%lon(1) = -180.0 + dlon/2.0
grid%lat(1) = -90.0 + dlat/2.0

grid%lon_bounds(1,1) = grid%lon(1) - dlon/2.0
grid%lon_bounds(2,1) = grid%lon(1) + dlon/2.0

grid%lat_bounds(1,1) = grid%lat(1) - dlat/2.0
grid%lat_bounds(2,1) = grid%lat(1) + dlat/2.0

DO i = 2, grid%nlon
  grid%lon(i) = grid%lon(i-1) + dlon
  grid%lon_bounds(1,i) = grid%lon(i) - dlon/2.0
  grid%lon_bounds(2,i) = grid%lon(i) + dlon/2.0
ENDDO

DO i = 2, grid%nlat
  grid%lat(i) = grid%lat(i-1) + dlat
  grid%lat_bounds(1,i) = grid%lat(i) - dlat/2.0
  grid%lat_bounds(2,i) = grid%lat(i) + dlat/2.0
ENDDO

! Get the normalized area weights corresponding to the grid.
CALL areaweights(grid%lon_bounds, grid%lat_bounds, grid%area)

WRITE(*,fmt='(a)')'Done.'

END SUBROUTINE gen_grid

!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!

SUBROUTINE areaweights(lon_bounds, lat_bounds, area)
! Purpose:
! This subroutine computes normalized surface area (weights) based on a
! set of latitude/longitude bounds.
!
! Record of revisions
! Date           Programmer      Description of change
! -----
! May 10, 2010   Chad Herman      Original code
! July 10, 2010  Chad Herman      Added USE numeric_kinds.
! Replaced KIND=r8b with KIND=r8b. Replaced KIND=lg with KIND=i4b.
! Replaced eps_r8b with eps_r8b. Replaced deg2rad_r8b with deg2rad_r8b.
! Replaced pi_d with pi_r8b.

USE numeric_kinds
USE constants

IMPLICIT NONE

```

```

REAL(KIND=r8b), DIMENSION(:,,:), INTENT(in) :: lon_bounds
REAL(KIND=r8b), DIMENSION(:,,:), INTENT(in) :: lat_bounds
REAL(KIND=r8b), DIMENSION(:,,:), INTENT(out) :: area

INTEGER(KIND=i4b) :: i, j

! Loop over the lat/lon bounds calculating the area weights. At this point,
! the figures calculated are not exactly correct. The lon_bounds variable is
! still in degrees.
DO j = 1, SIZE(lat_bounds,2)
  DO i = 1, SIZE(lon_bounds,2)
    area(i,j) = (sin(deg2rad_r8b*lat_bounds(1,j)) - sin(deg2rad_r8b*lat_bounds(2,j))) * &
& (lon_bounds(2,i) - lon_bounds(1,i))
  ENDDO
ENDDO

! Multiply by the degrees-to-radians conversion factor. Then divide by
! 4PI to normalize the area to a unit sphere.
area = deg2rad_r8b*ABS(area)/(4.0*pi_r8b)

END SUBROUTINE areaweights

!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!

SUBROUTINE gen_grid_index
! Purpose:
! This subroutine determines which stations are in which grid boxes.
!
! Record of revisions
! Date           Programmer           Description of change
! -----
! May 10, 2010   Chad Herman           Original code
! May 21, 2010   Chad Herman           Added .AND. to match_lon and match_lat
! to filter out the stations that aren't present in the station data file.
! July 10, 2010  Chad Herman           Replaced USE constants with
! USE numeric_kinds. Replaced KIND=lg with KIND=i4b.

USE numeric_kinds
USE ghcn_data, ONLY: stations, grid_index, grid

LOGICAL(KIND=i4b), DIMENSION(SIZE(stations)) :: match_lat, match_lon, match
INTEGER(KIND=i2b), DIMENSION(SIZE(stations)) :: index
INTEGER(KIND=i2b) :: i, j, err

WRITE(UNIT = *, FMT = '(a)', ADVANCE = 'no') 'Generating grid indices ... '

index = ((i,i=1,SIZE(stations))/)
ALLOCATE(grid_index(grid%nlon,grid%nlat,SIZE(stations)), STAT = err)
IF(err /= 0) THEN
  WRITE(*, FMT = '(A)') 'Unable to allocate memory for grid_index.'
  STOP
ENDIF

! Set all the values in the grid_index to missing which is -1.
grid_index = -1

! Loop over all the grid boxes and look for stations whose latitude/longitude coordinates
! are within the given grid box bounds.
DO j = 1, grid%nlat
  DO i = 1, grid%nlon

    match_lat = stations%lat > grid%lat_bounds(1,j) .AND. stations%lat < grid%lat_bounds(2,j) .AND. stations%index_0 /=
-1 match_lon = stations%lon > grid%lon_bounds(1,i) .AND. stations%lon < grid%lon_bounds(2,i) .AND. stations%index_0 /=
-1 match = match_lat .AND. match_lon

    IF(COUNT(match) > 0) THEN
      grid_index(i,j,1:COUNT(match)) = PACK(index, MASK = match)
    ENDIF

  ENDDO
ENDDO

WRITE(UNIT = *, FMT = '(a)') 'Done.'

END SUBROUTINE gen_grid_index

!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!

SUBROUTINE grid_data
! Purpose:
! This subroutine takes the station data after all its duplicates have
! been consolidated and combines all the stations in each grid box.
!
! Record of revisions
! Date           Programmer           Description of change
! -----
! May 11, 2010   Chad Herman           Original code
! May 21, 2010   Chad Herman           Added a mask to the year_0 and
! year_f calculation to filter out stations that don't have any data.
! July 10, 2010  Chad Herman           Added USE numeric_kinds.
! Replaced KIND=r8b with KIND=r8b. Replaced eps_r8b with eps_r8b.

USE constants
USE numeric_kinds
USE ghcn_parameters, ONLY: missing_dec
USE ghcn_data, ONLY: grid, gridded_data, stations, grid_index
USE ghcn_stat_comb

IMPLICIT NONE

```

```

INTEGER(KIND=i4b) :: nmonths, nstat_in_box, station_index
INTEGER(KIND=i4b), ALLOCATABLE, DIMENSION(:) :: number_of_stations
INTEGER(KIND=i4b) :: i, j, k, err
INTEGER(KIND=i4b) :: index_0, index_f
INTEGER(KIND=i4b) :: year_0, year_f
INTEGER(KIND=i4b), ALLOCATABLE, DIMENSION(:) :: indices
REAL(KIND=r8b), ALLOCATABLE, DIMENSION(:,) :: temp_data

WRITE(UNIT = *, FMT = '(a)', ADVANCE = 'no') 'Gridding data ... '

! Get the minimum and maximum temporal extent of the data and from that
! calculate how many monthly time steps there are.
year_0 = MINVAL(stations%year_0, MASK = stations%index_0 /= -1)
year_f = MAXVAL(stations%year_f, MASK = stations%index_0 /= -1)
nmonths = 12*(year_f - year_0 + 1)

! Write the temporal extent to the gridded variable data so the netCDF writer
! knows when the data begins and ends.
gridded_data%start_year = year_0
gridded_data%end_year = year_f

! Allocate memory for the indices variable to enable to indexing of January-only,
! February-only, etc. values.
ALLOCATE(indices(nmonths/12), STAT = err)
IF(err /= 0) THEN
  WRITE(UNIT = *, FMT = '(a)') 'Unable to allocate memory for indices.'
  STOP
ENDIF

! Allocate memory for the gridded data.
ALLOCATE(gridded_data%temperature(grid%nlon, grid%nlat, nmonths), STAT = err)
IF(err /= 0) THEN
  WRITE(UNIT = *, FMT = '(a)') 'Unable to allocate memory for gridded_data%temperature.'
  STOP
ENDIF

! Allocate memory for the number_of_stations variable in gridded_data.
ALLOCATE(gridded_data%number_of_stations(grid%nlon, grid%nlat, nmonths), STAT = err)
IF(err /= 0) THEN
  WRITE(UNIT = *, FMT = '(a)') 'Unable to allocate memory for gridded_data%number_of_stations.'
  STOP
ENDIF

! Allocate memory for the number of stations.
ALLOCATE(number_of_stations(nmonths), STAT = err)
IF(err /= 0) THEN
  WRITE(UNIT = *, FMT = '(a)') 'Unable to allocate memory for number_of_stations.'
  STOP
ENDIF

gridded_data%temperature = missing_dec
gridded_data%number_of_stations = 0

! Loop over all the grid boxes.
DO j = 1, grid%nlat
  DO i = 1, grid%nlon

! Get the number of stations in the grid box. If it's greater than zero, then proceed.
nstat_in_box = COUNT(grid_index(i, j, :)) /= -1

  IF(nstat_in_box > 0) THEN

! Allocate memory for the array to contain the station data.
ALLOCATE(temp_data(nmonths, nstat_in_box), STAT = err)
IF(err /= 0) THEN
  WRITE(UNIT = *, FMT = '(a)') 'Unable to allocate memory for temp_data.'
  STOP
ENDIF

! Initialize the allocated array with missing values.
temp_data = missing_dec

! Begin copying the station data into the array. Use the overall start and end year
! and the given station's start and end year to calculate indices so all the series
! will be temporally aligned.
DO k = 1, nstat_in_box
  station_index = grid_index(i, j, k)
  index_0 = 12*(stations(station_index)%year_0 - year_0) + 1
  index_f = 12*(stations(station_index)%year_f - year_0 + 1)
  temp_data(index_0:index_f, k) = stations(station_index)%temperature
ENDDO

! Get the number of stations reporting in the grid box for each time step.
number_of_stations = COUNT(ABS(temp_data - missing_dec) > eps_r8b, DIM = 2)
gridded_data%number_of_stations(i, j, :) = number_of_stations

! Now combine the stations together. If there is only one station, then skip the
! combination process.
IF(nstat_in_box == 1) THEN
  gridded_data%temperature(i, j, :) = temp_data(:, 1)
ELSE
  DO k = 1, 12
    indices = seq(k, nmonths, 12)
    gridded_data%temperature(i, j, indices) = combine_stations(temp_data(indices, :))
  ENDDO
ENDIF

  DEALLOCATE(temp_data)

ENDIF

```

```

        ENDDO
    ENDDO

    WRITE(UNIT = *, FMT = '(a)')'Done.'

END SUBROUTINE grid_data

!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!

END MODULE ghcn_grid_func

ANOMALY
MODULE ghcn_anomaly
CONTAINS

!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!

SUBROUTINE get_anomaly
! Purpose:
! This subroutine takes the gridded temperature and calculates the
! climatology at each grid point and removes it from the temperature to
! get the temperature anomaly.
!
! Record of revisions
! Date           Programmer           Description of change
! -----
! May 11, 2010   Chad Herman           Original code
! May 22, 2010   Chad Herman           Temperature variable is deallocated
! because it won't be saved to netCDF. This will also increase available
! memory which may be important when the anomalies and number of stations
! are packed into 2-byte integers.
! July 10, 2010  Chad Herman           Added USE numeric_kinds.
! Replaced KIND=r8b with KIND=r8b. Replaced KIND=lg with KIND=i4b.
! Replaced eps_r8b with eps_r8b.

USE constants
USE numeric_kinds
USE ghcn_data, ONLY: gridded_data
USE ghcn_parameters, ONLY: year0_b, yearf_b, min_n, missing_dec
USE ghcn_stat_comb, ONLY: seq

IMPLICIT NONE

REAL(KIND=r8b), DIMENSION(yearf_b - year0_b + 1) :: temperatures
LOGICAL(KIND=i4b), DIMENSION(yearf_b - year0_b + 1) :: real_temperatures
REAL(KIND=r8b), DIMENSION(12) :: climatology
REAL(KIND=r8b), DIMENSION(SIZE(gridded_data%temperature,3)) :: climatology_all, anomaly

INTEGER(KIND=i4b) :: i, j, k
INTEGER(KIND=i4b) :: nlon, nlat, ntime, nreal
INTEGER(KIND=i4b) :: index0, indexf
INTEGER(KIND=i4b) :: err

! Get the size of the incoming temperature data.
nlon = SIZE(gridded_data%temperature,1)
nlat = SIZE(gridded_data%temperature,2)
ntime = SIZE(gridded_data%temperature,3)

ALLOCATE(gridded_data%anomaly(nlon,nlat,ntime), stat = err)
IF(err /= 0)THEN
    WRITE(UNIT = *, FMT = '(a)')'Unable to allocate memory for gridded_data%anomaly.'
    STOP
ENDIF

! Get the first and last index along the time dimension
! that covers the base period.
index0 = 12*(year0_b - gridded_data%start_year) + 1
indexf = 12*(yearf_b - gridded_data%start_year + 1)

WRITE(UNIT = *, FMT = '(A)', ADVANCE = 'NO')'Calculating anomalies ... '

DO j = 1, nlat
    DO i = 1, nlon

        DO k = 1, 12

! Load the data for the (i,j) gridpoint. Calculate how many data points are
! not missing. If there's at least min_n, then calculate the average temperature. If
! not, then flag it as missing.
            temperatures = gridded_data%temperature(i,j,seq(index0 + k - 1, indexf, 12))
            real_temperatures = ABS(temperatures - missing_dec) > eps_r8b
            nreal = COUNT(real_temperatures)
            IF(nreal >= min_n)THEN
                climatology(k) = SUM(temperatures, MASK = real_temperatures)/nreal
            ELSE
                climatology(k) = missing_dec
            ENDIF

        ENDDO

! Now replicate the climatology into an array with the same size as the time
! dimension of the incoming temperature data.
        climatology_all = RESHAPE(climatology, SHAPE = (/ntime/), PAD = climatology)

! Get the anomaly.
        anomaly = gridded_data%temperature(i,j,:) - climatology_all

! It's possible that the entire annual cycle was not calculated. Where ever there

```

```

! are missing values in the original data OR missing values in the climatology, then
! flag the corresponding time steps to missing.

      WHERE(ABS(gridded_data%temperature(i,j,:) - missing_dec) < eps_r8b .OR. &
&      ABS(climatology_all - missing_dec) < eps_r8b) anomaly = missing_dec

! Now copy the result over the original data.
      gridded_data%anomaly(i,j,:) = anomaly

      ENDDO
ENDDO

! Deallocate the temperature data. We won't save it so it's just taking up scarce memory.
DEALLOCATE(gridded_data%temperature, stat = err)

WRITE(UNIT = *, FMT = '(A)') 'Done.'

END SUBROUTINE get_anomaly

!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!

END MODULE ghcn_anomaly

GHCN_WRITE
MODULE ghcn_write
CONTAINS
!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!

SUBROUTINE check_nc_op(status)
! Purpose:
! Checks to see if the netCDF operation was successful.
!
! Record of revisions
! Date          Programmer      Description of change
! -----
! May 11, 2010   Chad Herman    Original code
! July 10, 2010 Chad Herman    Replaced USE constants with
! USE numeric_kinds.

USE numeric_kinds
USE netcdf

INTEGER(KIND=i4b), INTENT(IN) :: status

IF(status /= nf90_noerr)THEN
  WRITE(UNIT = *, FMT = '(A)') TRIM(nf90_strerror(status))
  STOP
ENDIF

END SUBROUTINE check_nc_op

!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!

SUBROUTINE write_netCDF
! Purpose:
! Copies the gridded data variables to a netCDF file.
!
! Record of revisions
! Date          Programmer      Description of change
! -----
! May 11, 2010   Chad Herman    Original code
! May 22, 2010   Chad Herman    Dropped temperature variable. The
! remaining gridded variables are saved as short integers (16-bit). The
! anomaly is packed with an add offset and scale factor to recover the
! floating point representation.
! July 10, 2010   Chad Herman    Added USE numeric_kinds. Replaced
! eps_r4b with eps_r8b. Replaced_r4b with_r4b.

USE numeric_kinds
USE constants
USE ghcn_data, ONLY: gridded_data, grid
USE ghcn_parameters, ONLY: missing_dec, year0_b, yearf_b, min_n,&
                           stat_list_file, stat_data_file, outfile
USE netcdf

IMPLICIT NONE

CHARACTER(LEN=10) :: date

INTEGER(KIND=i4b) :: ncid
INTEGER(KIND=i4b) :: londimid, latdimid, timedimid, boundsdimid
INTEGER(KIND=i4b) :: lonvarid, latvarid, timevarid, lon_bndsvarid, lat_bndsvarid
INTEGER(KIND=i4b) :: anomalyvarid, numstatvarid, areavarid
INTEGER(KIND=i4b), ALLOCATABLE, DIMENSION(:) :: time

REAL(KIND=r8b) :: dataMin, dataMax, add_offset, scale_factor
INTEGER(KIND=i4b), PARAMETER :: n_bits = 16
INTEGER(KIND=i2b), DIMENSION(SIZE(gridded_data%anomaly,1),&
& SIZE(gridded_data%anomaly,2),&
& SIZE(gridded_data%anomaly,3)) :: packed_anomaly

INTEGER(KIND=i4b) :: start_year, end_year, nyears, nmonths
INTEGER(KIND=i4b) :: i, j, k
INTEGER(KIND=i4b) :: err

WRITE(UNIT = *, FMT = '(A)', ADVANCE = 'no') 'Writing data to netCDF ... '

! Get the date to write into the file.
CALL DATE_AND_TIME(date)

```



```

date = date(1:4) // '-' // date(5:6) // '-' // date(7:8)

! Get the start and end year to calculate create a vector of time values
! in the form yearmonth (201001, 201002, 201003, etc.)
start_year = gridded_data%start_year
end_year = gridded_data%end_year
nyears = end_year - start_year + 1
nmonths = 12*nyears

! Make space for the time variable.
ALLOCATE(time(nmonths), stat = err)

! Generate the time values.
k = 1
DO i = 1, nyears
  DO j = 1, 12
    time(k) = 100*(start_year + i - 1) + j
    k = k + 1
  ENDDO
ENDDO

! Get the minimum and maximum values for the anomaly data.
dataMin = MINVAL(gridded_data%anomaly, MASK = ABS(gridded_data%anomaly - missing_dec*1.0_r4b) > eps_r4b)
dataMax = MAXVAL(gridded_data%anomaly, MASK = ABS(gridded_data%anomaly - missing_dec*1.0_r4b) > eps_r4b)

! Calculate the scale_factor and add_offset for the anomaly data.
scale_factor = (dataMax - dataMin)/(2**n_bits - 1)
add_offset = dataMin + (2**(n_bits - 1))*scale_factor

! Initialize the packed_anomaly variable with the missing value.
packed_anomaly = missing_dec

! Pack the non-missing anomaly values into packed_anomaly.
WHERE(ABS(gridded_data%anomaly - missing_dec*1.0_r4b) > eps_r4b)&
&
  packed_anomaly = NINT((gridded_data%anomaly - add_offset)/scale_factor)

! Create the netCDF file.
CALL check_nc_op(nf90_create(path = TRIM(outfile), cmode = nf90_clobber, ncid = ncid))

! Define the longitude/latitude/time/bnds dimensions.
CALL check_nc_op(nf90_def_dim(ncid, "lon", grid%nlon, londimid))
CALL check_nc_op(nf90_def_dim(ncid, "lat", grid%nlat, latdimid))
CALL check_nc_op(nf90_def_dim(ncid, "time", nf90_unlimited, timedimid))
CALL check_nc_op(nf90_def_dim(ncid, "bnds", 2, boundsdimid))

! Define the longitude/latitude variables and their bounds.
CALL check_nc_op(nf90_def_var(ncid, "lon", nf90_double, londimid, lonvarid))
CALL check_nc_op(nf90_put_att(ncid, lonvarid, 'long_name', 'longitude'))
CALL check_nc_op(nf90_put_att(ncid, lonvarid, 'standard_name', 'longitude'))
CALL check_nc_op(nf90_put_att(ncid, lonvarid, 'units', 'degrees_east'))
CALL check_nc_op(nf90_put_att(ncid, lonvarid, 'bounds', 'lon_bnds'))

CALL check_nc_op(nf90_def_var(ncid, "lon_bnds", nf90_double, (/boundsdimid, londimid/), lon_bndsvarid))
CALL check_nc_op(nf90_put_att(ncid, lon_bndsvarid, 'long_name', 'longitude bounds'))
CALL check_nc_op(nf90_put_att(ncid, lon_bndsvarid, 'standard_name', 'longitude bounds'))
CALL check_nc_op(nf90_put_att(ncid, lon_bndsvarid, 'units', 'degrees_east'))

CALL check_nc_op(nf90_def_var(ncid, "lat", nf90_double, latdimid, latvarid))
CALL check_nc_op(nf90_put_att(ncid, latvarid, 'long_name', 'latitude'))
CALL check_nc_op(nf90_put_att(ncid, latvarid, 'standard_name', 'latitude'))
CALL check_nc_op(nf90_put_att(ncid, latvarid, 'units', 'degrees_north'))
CALL check_nc_op(nf90_put_att(ncid, latvarid, 'bounds', 'lat_bnds'))

CALL check_nc_op(nf90_def_var(ncid, "lat_bnds", nf90_double, (/boundsdimid, latdimid/), lat_bndsvarid))
CALL check_nc_op(nf90_put_att(ncid, lat_bndsvarid, 'long_name', 'latitude bounds'))
CALL check_nc_op(nf90_put_att(ncid, lat_bndsvarid, 'standard_name', 'latitude bounds'))
CALL check_nc_op(nf90_put_att(ncid, lat_bndsvarid, 'units', 'degrees_north'))

! Define the time variable.
CALL check_nc_op(nf90_def_var(ncid, "time", nf90_int, timedimid, timevarid))
CALL check_nc_op(nf90_put_att(ncid, timevarid, 'long_name', 'time'))
CALL check_nc_op(nf90_put_att(ncid, timevarid, 'standard_name', 'time'))
CALL check_nc_op(nf90_put_att(ncid, timevarid, 'units', 'month as %Y%m'))

! Define the anomaly variable and its attributes.
CALL check_nc_op(nf90_def_var(ncid, 'anomaly', nf90_short, (/londimid, latdimid, timedimid/), anomalyvarid))
CALL check_nc_op(nf90_put_att(ncid, anomalyvarid, 'long_name', 'anomaly'))
CALL check_nc_op(nf90_put_att(ncid, anomalyvarid, 'standard_name', 'anomaly'))
CALL check_nc_op(nf90_put_att(ncid, anomalyvarid, 'original_units', 'degC'))
CALL check_nc_op(nf90_put_att(ncid, anomalyvarid, 'units', 'degC'))
CALL check_nc_op(nf90_put_att(ncid, anomalyvarid, 'add_offset', add_offset))
CALL check_nc_op(nf90_put_att(ncid, anomalyvarid, 'scale_factor', scale_factor))
CALL check_nc_op(nf90_put_att(ncid, anomalyvarid, 'missing_value', INT(missing_dec, KIND=i2b)))
CALL check_nc_op(nf90_put_att(ncid, anomalyvarid, '_FillValue', INT(missing_dec, KIND=i2b)))

! Define the number_of_stations variable and its attributes.
CALL check_nc_op(nf90_def_var(ncid, 'number_of_stations', nf90_short, (/londimid, latdimid, timedimid/), numstatvarid))
CALL check_nc_op(nf90_put_att(ncid, numstatvarid, 'long_name', 'number of stations'))
CALL check_nc_op(nf90_put_att(ncid, numstatvarid, 'standard_name', 'number of stations'))

! Define the surface_area variable and its attributes.
CALL check_nc_op(nf90_def_var(ncid, 'area', nf90_double, (/londimid, latdimid/), areavarid))
CALL check_nc_op(nf90_put_att(ncid, areavarid, 'long_name', 'normalized surface area'))
CALL check_nc_op(nf90_put_att(ncid, areavarid, 'standard_name', 'surface area'))
CALL check_nc_op(nf90_put_att(ncid, areavarid, 'original_units', 'normalized to unit sphere'))
CALL check_nc_op(nf90_put_att(ncid, areavarid, 'units', 'normalized to unit sphere'))

! Define global attributes.
CALL check_nc_op(nf90_put_att(ncid, nf90_global, 'Title', 'Derived from GHCN Air Temperature Data'))
CALL check_nc_op(nf90_put_att(ncid, nf90_global, 'History', 'File created ' // date))
CALL check_nc_op(nf90_put_att(ncid, nf90_global, 'Base_period_start', year0_b))

```

```

CALL check_nc_op(nf90_put_att(ncid, nf90_global, 'Base_period_end', yearf_b))
CALL check_nc_op(nf90_put_att(ncid, nf90_global, 'Minimum_number_of_years_to_calculate_climatology', min_n))
CALL check_nc_op(nf90_put_att(ncid, nf90_global, 'Station_inventory_file', TRIM(stat_list_file)))
CALL check_nc_op(nf90_put_att(ncid, nf90_global, 'Station_data_file', TRIM(stat_data_file)))
CALL check_nc_op(nf90_put_att(ncid, nf90_global, 'Conventions', 'Based on CF-1.4'))

! Exit define mode.
CALL check_nc_op(nf90_enddef(ncid))

! Write the latitude, longitude and time variables.
CALL check_nc_op(nf90_put_var(ncid, latvarid, grid%lat))
CALL check_nc_op(nf90_put_var(ncid, lonvarid, grid%lon))
CALL check_nc_op(nf90_put_var(ncid, timevarid, time))

! Write the latitude/longitude bounds variables.
CALL check_nc_op(nf90_put_var(ncid, lon_bndsvarid, grid%lon_bounds))
CALL check_nc_op(nf90_put_var(ncid, lat_bndsvarid, grid%lat_bounds))

! Write the temperature, anomaly, number of stations and surface area variables.
! CALL check_nc_op(nf90_put_var(ncid, tempvarid, gridded_data%temperature))
CALL check_nc_op(nf90_put_var(ncid, anomalyvarid, packed_anomaly))
CALL check_nc_op(nf90_put_var(ncid, numstatvarid, gridded_data%number_of_stations))
CALL check_nc_op(nf90_put_var(ncid, areavarid, grid%area))

! Close the netCDF file.
CALL check_nc_op(nf90_close(ncid))

WRITE(UNIT = *, FMT = '(A)') 'Done.'

END SUBROUTINE write_netCDF

!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!

END MODULE ghcn_write

GET_PARAMETERS
MODULE ghcn_parameters
! Purpose:
! This module contains definitions of the data format for the station
! inventory list, the raw station data, missing data values for the
! original and assimilated data, the grid box size, the temporal
! range for the calculated climatology and the minimum number of real
! valued data points for calculating a valid climatology.
!
! Record of revisions
! Date           Programmer           Description of change
! -----
! May 10, 2010   Chad Herman           Original code
! May 21, 2010   Chad Herman           Added al to the end of the
! stat_list_fmt for 'brightness'.
! May 24, 2010   Chad Herman           Added start_year and end_year.
! July 10, 2010  Chad Herman           Replaced USE constants with
! USE numeric_kinds. Changed KIND=dp to KIND=r8b. Replaced _dp with _r8b.

USE numeric_kinds

!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!

CHARACTER(LEN=*), PARAMETER :: parameter_file = 'ghcn_parameters.dat'

CHARACTER(LEN=*), PARAMETER :: stat_list_fmt = '(i3.3,i5.5,i3.3,1x,a30,&
&1x,f6.2,1x,f7.2,1x,i4,1x,i4,a1,i5,3(a2),i2,a1,i2,a16,a1)'

CHARACTER(LEN=*), PARAMETER :: stat_data_fmt = '(i3,i5,i3,i1,i4,12i5)'
CHARACTER(LEN=*), PARAMETER :: stat_data_fmt_year = '(12x,i4)'

INTEGER(KIND=i4b), PARAMETER :: ghcn_missing = -9999
INTEGER(KIND=i4b), PARAMETER :: missing_int = -9990
REAL(KIND=r8b), PARAMETER :: missing_dec = -999.0_r8b

CHARACTER(LEN=256) :: stat_list_file
CHARACTER(LEN=256) :: stat_data_file
REAL(KIND=r8b) :: dlat
REAL(KIND=r8b) :: dlon
CHARACTER(LEN=256) :: outfile
INTEGER(KIND=i4b) :: year0_b
INTEGER(KIND=i4b) :: yearf_b
INTEGER(KIND=i4b) :: min_n
INTEGER(KIND=i4b) :: start_year
INTEGER(KIND=i4b) :: end_year
INTEGER(KIND=i4b) :: nc_cmode

!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!

END MODULE ghcn_parameters

```

Figure 1-3: Station Count

```

rm(list = ls())

library(RNetCDF)

file = 'ghcn_bp_1961-1990_5x5_v2.mean.nc'
adj.file = 'ghcn_adj_bp_1961-1990_5x5_v2.mean.nc'

nc = open.nc(file, write = FALSE)

time = var.get.nc(nc, 'time'); ntime = length(time)
lon = var.get.nc(nc, 'lon'); nlon = length(lon)

```

```

lat = var.get.nc(nc, 'lat'); nlat = length(lat)
num_stat = var.get.nc(nc, 'number_of_stations')

close.nc(nc)

nc = open.nc(adj.file, write = FALSE)
num_stat.adj = var.get.nc(nc, 'number_of_stations')
close.nc(nc)

nstat = matrix(NA, ntime, 3); colnames(avg) = c("GLOBAL", "NH", "SH")
nstat.adj = nstat

north = which(lat >= 0)
south = which(lat <= 0)

time.steps = cbind( floor(time/100), time - 100*floor(time/100) )

for(i in 1:ntime){
  nstat[i,1] = sum(num_stat[,i], na.rm = T)
  nstat[i,2] = sum(num_stat[,north,i], na.rm = T)
  nstat[i,3] = sum(num_stat[,south,i], na.rm = T)
}

for(i in 1:ntime){
  nstat.adj[i,1] = sum(num_stat.adj[,i], na.rm = T)
  nstat.adj[i,2] = sum(num_stat.adj[,north,i], na.rm = T)
  nstat.adj[i,3] = sum(num_stat.adj[,south,i], na.rm = T)
}

out.file = "ghcn_stat_count.dat"
out.file.adj = "ghcn_stat_count_adj.dat"
colnames = sprintf("%4s %5s %6s %6s %6s", "YEAR", "MONTH", "GLOBAL", "NH", "SH")

header = rbind( c("GHCN Station Count"),
                c("")
              )
data = sprintf("%4.0f %5.0f %6.0f %6.0f %6.0f", time.steps[1], time.steps[2], nstat[1,1], nstat[1,2], nstat[1,3])
write(header, file = out.file)
write(colnames, file = out.file, append = TRUE)
write(data, file = out.file, append = TRUE)

header = rbind( c("GHCN Station Count (Adjusted)"),
                c("")
              )
data = sprintf("%4.0f %5.0f %6.0f %6.0f %6.0f", time.steps[1], time.steps[2], nstat.adj[1,1], nstat.adj[1,2],
              nstat.adj[1,3])
write(header, file = out.file.adj)
write(colnames, file = out.file.adj, append = TRUE)
write(data, file = out.file.adj, append = TRUE)

```

Figure 1-6: Airport count

```

PROGRAM ghcn_airport_stat_count

USE numeric_kinds
USE constants
USE ghcn_import, ONLY: get_parameters, get_stations, load_station_data, get_station_indices
USE ghcn_stat_comb, ONLY: consolidate_duplicates
USE ghcn_data, ONLY : stations
USE ghcn_parameters

IMPLICIT NONE

INTEGER(KIND=i4b) :: i, j, k
INTEGER(KIND=i4b), ALLOCATABLE, DIMENSION(:,:) :: north, south, global, time
INTEGER(KIND=i4b), ALLOCATABLE, DIMENSION(:) :: temp
REAL(KIND=r8b), ALLOCATABLE, DIMENSION(:) :: values
INTEGER(KIND=i4b) :: ntime
INTEGER(KIND=i4b) :: err
INTEGER(KIND=i4b) :: index_0, index_f

CALL get_parameters
CALL get_stations
CALL load_station_data
CALL get_station_indices
CALL consolidate_duplicates

ntime = 12*(end_year - start_year + 1)

! north = (/ # of airport, # of non-airport /)
ALLOCATE(north(ntime,2), south(ntime,2), global(ntime,2), temp(ntime), time(ntime,2), values(ntime), STAT = err)
IF(err /= 0)THEN
  WRITE(UNIT = 1, FMT = "(A)") "Unable to allocate memory for the count/time variables."
  STOP
ENDIF

north = 0
south = 0
global = 0
temp = 0

! Generate time variable.
k = 1
DO i = start_year, end_year

```

```

DO j = 1, 12
  time(k,:) = (/i,j/)
  k = k + 1
ENDDO
ENDDO

DO i = 1, SIZE(stations)

  IF (ALLOCATED(stations(i)%temperature)) THEN

! Copy the station data into the values vector and align it.
  values = missing_dec
  index_0 = 12*(stations(i)%year_0 - start_year) + 1
  index_f = 12*(stations(i)%year_f - start_year + 1)
  values(index_0:index_f) = stations(i)%temperature

! Set temp to one for non-missing data and zero for missing data.
  WHERE( ABS(values - missing_dec) > eps_r8b )
    temp = 1
  ELSEWHERE
    temp = 0
  ENDWHERE

! Check which hemisphere the station is in. Then determine if it's at
! an airport.
  IF(stations(i)%lat > 0.0) THEN

    IF(stations(i)%airport == "A") THEN
      north(:,1) = north(:,1) + temp
    ELSE
      north(:,2) = north(:,2) + temp
    ENDIF

  ELSE

    IF(stations(i)%airport == "A") THEN
      south(:,1) = south(:,1) + temp
    ELSE
      south(:,2) = south(:,2) + temp
    ENDIF

  ENDIF

ENDIF

ENDDO

! Combine the north and south.
global = north + south

! Write the results to a file.
OPEN(UNIT = 1, FILE = 'ghcn_airport_count.dat')
WRITE(UNIT = 1, FMT = '(A)') 'DATA: Number of GHCN stations at an airport.'
WRITE(UNIT = 1, FMT = '(A)') ''
WRITE(UNIT = 1, FMT = '(5(A6,1x))') 'YEAR', 'MONTH', 'GLOBAL', 'NH', 'SH'
DO i = 1, ntime
  WRITE(UNIT = 1, FMT = '(5(I6,1x))') time(i,:), global(i,1), north(i,1), south(i,1)
ENDDO
CLOSE(UNIT = 1)

OPEN(UNIT = 1, FILE = 'ghcn_non-airport_count.dat')
WRITE(UNIT = 1, FMT = '(A)') 'DATA: Number of GHCN stations not at an airport.'
WRITE(UNIT = 1, FMT = '(A)') ''
WRITE(UNIT = 1, FMT = '(5(A6,1x))') 'YEAR', 'MONTH', 'GLOBAL', 'NH', 'SH'
DO i = 1, ntime
  WRITE(UNIT = 1, FMT = '(5(I6,1x))') time(i,:), global(i,2), north(i,2), south(i,2)
ENDDO
CLOSE(UNIT = 1)

END PROGRAM ghcn_airport_stat_count

```

**(Stata code to draw graph)**

```

capture log close
log using airport.log, replace
clear

*      READ AIRPORT COUNT DATA
insheet airport using ghcn_airport_count.dat
      split airport                                // split columns, remove header and destring
      drop if _n==1 | _n==2
      drop airport
      destring, replace

      rename airport1 year                        // put proper names on variables
      rename airport2 month
      rename airport3 airport
      rename airport4 air_nh
      rename airport5 air_sh
      drop airport6-airport8

sum

save airport ,replace

clear

*      READ NON-AIRPORT COUNT DATA
insheet nonairport using ghcn_nonairport_count.dat
      split nonairport                            // split columns, remove header and destring
      drop if _n==1 | _n==2
      drop nonairport

```

```

destring, replace

rename nonairport1 year // put proper names on variables
rename nonairport2 month
rename nonairport3 nonairport
rename nonairport4 nonair_nh
rename nonairport5 nonair_sh
drop nonairport6-nonairport9

sum

save nonairport,replace

clear

* MERGE DATA & REMOVE ZEROS AT END
use airport
merge using nonairport
sum

drop if year==2010 & month>6
sum

* MAKE CHARTS
generate ghcnc =airport+nonairport

collapse ghcnc airport nonairport air_nh air_sh nonair_nh nonair_sh , by(year)
sum

generate pctairport=100*airport/ghcnc
generate nh_air=100*air_nh/(air_nh+nonair_nh)
generate sh_air=100*air_sh/(air_sh+nonair_sh)

generate pct_nh=(air_nh+nonair_nh)/(air_nh+nonair_nh+air_sh+nonair_sh)

list year pctairport nh_air sh_air if year>1980
exit

line ghcnc year , xlabel(1870(20)2010)
graph save ghcnc,replace
line pctairport year , xlabel(1890(40)2010) ylabel(0(20)80)
graph save pct_air, replace
line nh_air year , xlabel(1890(40)2010) ylabel(0(20)80)
graph save nh_air, replace
line sh_air year , xlabel(1890(40)2010) ylabel(0(20)80)
graph save sh_air, replace

graph combine pct_air.gph nh_air.gph sh_air.gph, r(1)
graph save airports, replace

exit

```

Figure 1-7: Latitude

```

rm(list=ls())

library(RNetCDF)

file = "noaa/ghcn_bp_1961-1990_5x5_v2.mean.nc"

nc = open.nc(file, write = FALSE)

lat = var.get.nc(nc, 'lat'); nlat = length(lat)
time = var.get.nc(nc, 'time'); ntime = length(time)
time.steps = cbind(floor(time/100), time - 100*floor(time/100))

num_stat = var.get.nc(nc, "number_of_stations")

close.nc(nc)

weights = cos(lat*pi/180)
weights = weights/sum(weights)

lat_mean_unweighted = rep(NA, ntime)
lat_mean_weighted = rep(NA, ntime)

lat = abs(lat)

for(i in 1:ntime){
  temp = colSums(num_stat[,i])
  if(sum(temp) != 0) lat_mean_unweighted[i] = sum(temp*lat)/sum(temp) else lat_mean_unweighted[i] = -999
  if(sum(temp) != 0) lat_mean_weighted[i] = sum(temp*lat*weights)/sum(temp*weights) else lat_mean_weighted[i] =
-999
}

out.file = "ghcn_mean_abs_latitude.dat"
colnames = sprintf("%4s %5s %10s %10s", "YEAR", "MONTH", "UNWEIGHTED", "WEIGHTED")

header = rbind( c("GHCN Mean Latitude"),
               c("")
               )

data = sprintf("%4.0f %5.0f %10.4f %10.4f", time.steps[,1], time.steps[,2], lat_mean_unweighted, lat_mean_weighted)

write(header, file = out.file)
write(colnames, file = out.file, append = TRUE)
write(data, file = out.file, append = TRUE)

lat_mean_unweighted[lat_mean_unweighted == -999] = NA
lat_mean_weighted[lat_mean_weighted == -999] = NA

lat_mean_unweighted = ts(lat_mean_unweighted, start = time.steps[1,], freq = 12)

```

```

lat_mean_weighted = ts(lat_mean_weighted, start = time.steps[1,], freq = 12)

png("ghcn_mean_abs_latitude_unweighted.png", bg="white", width=600, height=600)
plot(lat_mean_unweighted, xlab = 'Year', ylab = 'Latitude', main = 'GHCN Mean Latitude (Unweighted)', type = 'l')
dev.off()

png("ghcn_mean_abs_latitude_weighted.png", bg="white", width=600, height=600)
plot(lat_mean_weighted, xlab = 'Year', ylab = 'Latitude', main = 'GHCN Mean Latitude (Weighted)', type = 'l')
dev.off()

png("ghcn_mean_abs_latitude.png", bg="white", width=600, height=600)
plot(lat_mean_unweighted, xlab = 'Year', ylab = 'Latitude', main = 'GHCN Mean Latitude (Unweighted and Weighted)', type = 'l', col = 'red', ylim = range(c(lat_mean_unweighted, lat_mean_weighted), na.rm = T))
lines(lat_mean_weighted, col = 'blue')
legend(x="bottomleft", legend=c("Unweighted", "Weighted"), col=c("red", "blue"), lty=1, lwd=2)
dev.off()

```

Figure 1-8: Altitude

```

PROGRAM ghcn_mean_altitude

USE numeric_kinds
USE constants
USE ghcn_import, ONLY: get_parameters, get_stations, load_station_data, get_station_indices
USE ghcn_stat_comb, ONLY: consolidate_duplicates
USE ghcn_data, ONLY : stations
USE ghcn_parameters

IMPLICIT NONE

INTEGER(KIND=i4b) :: i, j, k
INTEGER(KIND=i4b), ALLOCATABLE, DIMENSION(:,:) :: time
INTEGER(KIND=i4b), ALLOCATABLE, DIMENSION(:) :: temp
REAL(KIND=r8b), ALLOCATABLE, DIMENSION(:) :: values
REAL(KIND=r8b), ALLOCATABLE, DIMENSION(:) :: alt, alt_int
INTEGER(KIND=i4b), ALLOCATABLE, DIMENSION(:) :: n_stat, n_stat_int
INTEGER(KIND=i4b) :: ntime
INTEGER(KIND=i4b) :: err
INTEGER(KIND=i4b) :: index_0, index_f

CALL get_parameters
CALL get_stations
CALL load_station_data
CALL get_station_indices
CALL consolidate_duplicates

ntime = 12*(end_year - start_year + 1)

ALLOCATE(time(ntime,2), values(ntime), temp(ntime), alt(ntime), alt_int(ntime), n_stat(ntime), n_stat_int(ntime), STAT = err)
IF(err /= 0)THEN
  WRITE(UNIT = 1, FMT = "(A)") "Unable to allocate memory for the count/time variables."
  STOP
ENDIF

! Generate time variable.
k = 1
DO i = start_year, end_year
  DO j = 1, 12
    time(k,:) = (/i,j/)
    k = k + 1
  ENDDO
ENDDO

alt = 0
alt_int = 0
n_stat = 0
n_stat_int = 0

DO i = 1, SIZE(stations)

  IF(ALLOCATED(stations(i)%temperature))THEN

! Copy the station data into the values vector and align it.
  values = missing_dec
  index_0 = 12*(stations(i)%year_0 - start_year) + 1
  index_f = 12*(stations(i)%year_f - start_year + 1)
  values(index_0:index_f) = stations(i)%temperature

! Set temp to one for non-missing data and zero for missing data.
  WHERE( ABS(values - missing_dec) > eps_r8b )
    temp = 1
  ELSEWHERE
    temp = 0
  ENDWHERE

  IF(stations(i)%elev /= -999)THEN
    n_stat = n_stat + temp
    alt = alt + stations(i)%elev*temp
  ENDIF

  n_stat_int = n_stat_int + temp
  alt_int = alt_int + stations(i)%elev_int*temp

ENDIF

ENDDO

! Get average altitude.
WHERE(n_stat /= 0)

```

```

    alt = alt/n_stat
ELSEWHERE
    alt = -999
ENDWHERE

WHERE(n_stat_int /= 0)
    alt_int = alt_int/n_stat_int
ELSEWHERE
    alt_int = -999
ENDWHERE

! Write the results to a file.
OPEN(UNIT = 1, FILE = 'ghcn_mean_altitude.dat')
WRITE(UNIT = 1, FMT = '(A)') 'DATA: Number of GHCN stations at an airport.'
WRITE(UNIT = 1, FMT = '(A)') ''
WRITE(UNIT = 1, FMT = '(A4,1X,A5,1X,2(A12,1X))') 'YEAR', 'MONTH', 'ALTITUDE', 'ALTITUDE_INT'
DO i = 1, ntime
    WRITE(UNIT = 1, FMT = '(I4,1X,I5,1X,2(F12.3,1X))') time(i,:), alt(i), alt_int(i)
ENDDO
CLOSE(UNIT = 1)

END PROGRAM ghcn_mean_altitude

```

Figure 1-10: GHCN ADJ

```

!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!
SUBROUTINE check_nc_op(status)

    USE numeric_kinds
    USE netcdf

    INTEGER(KIND=i4b), INTENT(IN) :: status

    IF(status /= nf90_noerr)THEN
        WRITE(UNIT = *, FMT = '(A)')TRIM(nf90_strerror(status))
        STOP
    ENDIF

END SUBROUTINE check_nc_op

!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!

PROGRAM mask_ghcn_obs

    USE numeric_kinds
    USE constants
    USE netcdf

    !!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!

    IMPLICIT NONE

    CHARACTER(LEN=*) , PARAMETER :: ghcn_file = "noaa/ghcn_bp_1961-1990_5x5_v2.mean.nc"
    CHARACTER(LEN=*) , PARAMETER :: ghcn_adj_file = "noaa/ghcn_adj_bp_1961-1990_5x5_v2.mean.nc"
    CHARACTER(LEN=*) , PARAMETER :: land_mask_file = "ssmi_land_mask.nc"

    INTEGER(KIND=i4b), PARAMETER :: nlon = 72, nlat = 36
    REAL(KIND=r8b), DIMENSION(2,nlon) :: lon_bnds
    REAL(KIND=r8b), DIMENSION(2,nlat) :: lat_bnds
    REAL(KIND=r8b), DIMENSION(nlon,nlat) :: area_weights

    INTEGER(KIND=i4b), PARAMETER :: nlon_land_mask = 1440, nlat_land_mask = 720
    REAL(KIND=r4b), DIMENSION(nlon_land_mask) :: lon_mask
    REAL(KIND=r4b), DIMENSION(nlat_land_mask) :: lat_mask
    REAL(KIND=r4b), DIMENSION(nlon_land_mask,nlat_land_mask) :: land_mask
    REAL(KIND=r4b), DIMENSION(nlon,nlat) :: land_mask_interp

    INTEGER(KIND=i4b), DIMENSION(nlon_land_mask) :: lon_mask_indices
    INTEGER(KIND=i4b), DIMENSION(nlat_land_mask) :: lat_mask_indices
    LOGICAL(KIND=i4b), DIMENSION(nlon_land_mask) :: lon_match
    LOGICAL(KIND=i4b), DIMENSION(nlat_land_mask) :: lat_match
    INTEGER(KIND=i4b) :: nlon_match, nlat_match
    REAL(KIND=r8b) :: max_land_area

    INTEGER(KIND=i4b) :: status, err
    INTEGER(KIND=i4b) :: ncid_ghcn, ncid_ghcn_adj
    INTEGER(KIND=i4b) :: time_dimid_ghcn, time_dimid_ghcn_adj
    INTEGER(KIND=i4b) :: time_varid_ghcn, time_varid_ghcn_adj
    INTEGER(KIND=i4b) :: anomaly_varid_ghcn, anomaly_varid_ghcn_adj
    INTEGER(KIND=i4b) :: lon_bnds_varid, lat_bnds_varid

    INTEGER(KIND=i4b) :: ncid
    INTEGER(KIND=i4b) :: lon_varid, lat_varid, land_mask_varid

    REAL(KIND=r4b) :: missing_ghcn, missing_ghcn_adj
    REAL(KIND=r4b) :: add_offset_ghcn, scale_factor_ghcn
    REAL(KIND=r4b) :: add_offset_ghcn_adj, scale_factor_ghcn_adj

    INTEGER(KIND=i4b) :: ntime_ghcn, ntime_ghcn_adj, ntime
    INTEGER(KIND=i4b), ALLOCATABLE, DIMENSION(:) :: time_ghcn, time_ghcn_adj
    INTEGER(KIND=i4b), ALLOCATABLE, DIMENSION(:,) :: time_ghcn_YM, time_ghcn_adj_YM

    REAL(KIND=i4b), ALLOCATABLE, DIMENSION(:,,:) :: anomaly_ghcn, anomaly_ghcn_adj
    LOGICAL(KIND=i4b), DIMENSION(nlon,nlat) :: mask, mask_ghcn, mask_ghcn_adj

    REAL(KIND=r4b), ALLOCATABLE, DIMENSION(:,) :: spatial_avg_common, spatial_avg
    REAL(KIND=r4b), ALLOCATABLE, DIMENSION(:) :: area_common
    REAL(KIND=r4b), ALLOCATABLE, DIMENSION(:,) :: area

```

```

REAL(KIND=r4b), PARAMETER :: missing = -999._r4b

INTEGER(KIND=i4b) :: year0, month0
INTEGER(KIND=i4b) :: yearf, monthf

INTEGER(KIND=i4b) :: ghcn_start, ghcn_end
INTEGER(KIND=i4b) :: ghcn_adj_start, ghcn_adj_end

INTEGER(KIND=i4b) :: i, j, k
!*****

WRITE(UNIT = *, FMT = '(A)', ADVANCE = 'NO') 'Opening the netCDF files ... '

status = nf90_open(ghcn_file, nf90_nowrite, ncid_ghcn)
IF(status /= nf90_noerr)THEN
  WRITE(UNIT = *, FMT = *)
  WRITE(UNIT = *, FMT = '(A)') 'Attempted to open ' // ghcn_file
  WRITE(UNIT = *, FMT = '(A)') TRIM(nf90_strerror(status))
  STOP
ENDIF

status = nf90_open(ghcn_adj_file, nf90_nowrite, ncid_ghcn_adj)
IF(status /= nf90_noerr)THEN
  WRITE(UNIT = *, FMT = *)
  WRITE(UNIT = *, FMT = '(A)') 'Attempted to open ' // ghcn_adj_file
  WRITE(UNIT = *, FMT = '(A)') TRIM(nf90_strerror(status))
  STOP
ENDIF

WRITE(UNIT = *, FMT = '(A)') 'Done.'

!*****

WRITE(UNIT = *, FMT = '(A)', ADVANCE = 'NO') 'Getting the time dimension/variable ids and their lengths ... '

CALL check_nc_op(nf90_inq_dimid(ncid_ghcn, 'time', time_dimid_ghcn))
CALL check_nc_op(nf90_inq_varid(ncid_ghcn, 'time', time_varid_ghcn))
CALL check_nc_op(nf90_inquire_dimension(ncid_ghcn, time_dimid_ghcn, len = ntime_ghcn))

CALL check_nc_op(nf90_inq_dimid(ncid_ghcn_adj, 'time', time_dimid_ghcn_adj))
CALL check_nc_op(nf90_inq_varid(ncid_ghcn_adj, 'time', time_varid_ghcn_adj))
CALL check_nc_op(nf90_inquire_dimension(ncid_ghcn_adj, time_dimid_ghcn_adj, len = ntime_ghcn_adj))

WRITE(UNIT = *, FMT = '(A)') 'Done.'

!*****

WRITE(UNIT = *, FMT = '(A)', ADVANCE = 'NO') 'Getting the lon/lat bounds variable ids ... '

CALL check_nc_op(nf90_inq_varid(ncid_ghcn, 'lon_bnds', lon_bnds_varid))
CALL check_nc_op(nf90_inq_varid(ncid_ghcn, 'lat_bnds', lat_bnds_varid))

WRITE(UNIT = *, FMT = '(A)') 'Done.'

!*****

WRITE(UNIT = *, FMT = '(A)', ADVANCE = 'NO') 'Getting the anomaly variable ids and their missing values ... '

CALL check_nc_op(nf90_inq_varid(ncid_ghcn, 'anomaly', anomaly_varid_ghcn))
CALL check_nc_op(nf90_get_att(ncid_ghcn, anomaly_varid_ghcn, 'missing_value', missing_ghcn))

CALL check_nc_op(nf90_inq_varid(ncid_ghcn_adj, 'anomaly', anomaly_varid_ghcn_adj))
CALL check_nc_op(nf90_get_att(ncid_ghcn_adj, anomaly_varid_ghcn_adj, 'missing_value', missing_ghcn_adj))

WRITE(UNIT = *, FMT = '(A)') 'Done.'

!*****

WRITE(UNIT = *, FMT = '(A)', ADVANCE = 'NO') 'Allocating memory for the time variables ... '

ALLOCATE(time_ghcn(ntime_ghcn), time_ghcn_adj(ntime_ghcn_adj), STAT = err)
IF(err /= 0)THEN
  WRITE(UNIT = *, FMT = *)
  WRITE(UNIT = *, FMT = *) 'Unable to allocate memory for the time variables.'
  STOP
ENDIF

ALLOCATE(time_ghcn_YM(ntime_ghcn,2), time_ghcn_adj_YM(ntime_ghcn_adj,2), STAT = err)
IF(err /= 0)THEN
  WRITE(UNIT = *, FMT = *)
  WRITE(UNIT = *, FMT = *) 'Unable to allocate memory for the time variables (year/month).'
  STOP
ENDIF

WRITE(UNIT = *, FMT = '(A)') 'Done.'

!*****

WRITE(UNIT = *, FMT = '(A)', ADVANCE = 'NO') 'Loading time variables ... '

CALL check_nc_op(nf90_get_var(ncid_ghcn, time_varid_ghcn, time_ghcn))
CALL check_nc_op(nf90_get_var(ncid_ghcn_adj, time_varid_ghcn_adj, time_ghcn_adj))

WRITE(UNIT = *, FMT = '(A)') 'Done.'

!*****

WRITE(UNIT = *, FMT = '(A)', ADVANCE = 'NO') 'Getting the longitude/latitude bounds ... '

CALL check_nc_op(nf90_get_var(ncid_ghcn, lon_bnds_varid, lon_bnds))
CALL check_nc_op(nf90_get_var(ncid_ghcn, lat_bnds_varid, lat_bnds))

```



```

WRITE(UNIT = *, FMT = '(A)') 'Done.'
!%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
WRITE(UNIT = *, FMT = '(A)', ADVANCE = 'NO') 'Breaking time variables into year/month ... '
time_ghcn_YM(:,1) = FLOOR(time_ghcn/100._r4b)
time_ghcn_YM(:,2) = time_ghcn - 100*time_ghcn_YM(:,1)
time_ghcn_adj_YM(:,1) = FLOOR(time_ghcn_adj/100._r4b)
time_ghcn_adj_YM(:,2) = time_ghcn_adj - 100*time_ghcn_adj_YM(:,1)
WRITE(UNIT = *, FMT = '(A)') 'Done.'
!%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
WRITE(UNIT = *, FMT = '(A)', ADVANCE = 'NO') 'Calculating the start/end periods and time offsets ... '
year0 = max(time_ghcn_YM(1,1), time_ghcn_adj_YM(1,1))
yearf = min(time_ghcn_YM(ntime_ghcn,1), time_ghcn_adj_YM(ntime_ghcn_adj,1))
month0 = max(time_ghcn_YM(1,2), time_ghcn_adj_YM(1,2))
monthf = min(time_ghcn_YM(ntime_ghcn,2), time_ghcn_adj_YM(ntime_ghcn_adj,2))
ghcn_start = 12*(year0 - time_ghcn_YM(1,1)) + 1 + (month0 - 1)
ghcn_end = 12*(yearf - time_ghcn_YM(1,1)) + 1 + (monthf - 1)
ghcn_adj_start = 12*(year0 - time_ghcn_adj_YM(1,1)) + 1 + (month0 - 1)
ghcn_adj_end = 12*(yearf - time_ghcn_adj_YM(1,1)) + 1 + (monthf - 1)
ntime = ghcn_end - ghcn_start + 1
WRITE(UNIT = *, FMT = '(A)') 'Done.'
!%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
WRITE(UNIT = *, FMT = '(A)', ADVANCE = 'NO') 'Allocating memory for the anomaly variables and mask ... '
ALLOCATE(anomaly_ghcn(nlon,nlat,ntime), anomaly_ghcn_adj(nlon,nlat,ntime), STAT = err)
IF(err /= 0) THEN
  WRITE(UNIT = *, FMT = *)
  WRITE(UNIT = *, FMT = *) 'Unable to allocate memory for the anomaly variables.'
  STOP
ENDIF
WRITE(UNIT = *, FMT = '(A)') 'Done.'
!%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
WRITE(UNIT = *, FMT = '(A)', ADVANCE = 'NO') 'Allocating memory for the spatial average/area variables ... '
ALLOCATE(spatial_avg_common(ntime,2), spatial_avg(ntime,2), area(ntime,2), area_common(ntime), STAT = err)
IF(err /= 0) THEN
  WRITE(UNIT = *, FMT = *)
  WRITE(UNIT = *, FMT = *) 'Unable to allocate memory for the spatial average variables.'
  STOP
ENDIF
WRITE(UNIT = *, FMT = '(A)') 'Done.'
!%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
WRITE(UNIT = *, FMT = '(A)', ADVANCE = 'NO') 'Loading anomaly variables ... '
CALL check_nc_op(nf90_get_var(ncid_ghcn, anomaly_varid_ghcn, anomaly_ghcn, &
& start = (/1,1,ghcn_start/), count = (/nlon,nlat,ntime/)))
CALL check_nc_op(nf90_get_var(ncid_ghcn_adj, anomaly_varid_ghcn_adj, anomaly_ghcn_adj, &
& start = (/1,1,ghcn_adj_start/), count = (/nlon,nlat,ntime/)))
WRITE(UNIT = *, FMT = '(A)') 'Done.'
!%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
WRITE(UNIT = *, FMT = '(A)', ADVANCE = 'NO') 'Loading anomaly add_offset and scale_factor values ... '
CALL check_nc_op(nf90_get_att(ncid_ghcn, anomaly_varid_ghcn, 'add_offset', add_offset_ghcn))
CALL check_nc_op(nf90_get_att(ncid_ghcn_adj, anomaly_varid_ghcn_adj, 'add_offset', add_offset_ghcn_adj))
CALL check_nc_op(nf90_get_att(ncid_ghcn, anomaly_varid_ghcn, 'scale_factor', scale_factor_ghcn))
CALL check_nc_op(nf90_get_att(ncid_ghcn_adj, anomaly_varid_ghcn_adj, 'scale_factor', scale_factor_ghcn_adj))
WRITE(UNIT = *, FMT = '(A)') 'Done.'
!%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
WRITE(UNIT = *, FMT = '(A)', ADVANCE = 'NO') 'Unpacking anomaly data ... '
WHERE( ABS(anomaly_ghcn - missing_ghcn) > eps_r4b ) anomaly_ghcn = add_offset_ghcn + scale_factor_ghcn*anomaly_ghcn
WHERE( ABS(anomaly_ghcn_adj - missing_ghcn_adj) > eps_r4b ) anomaly_ghcn_adj = add_offset_ghcn_adj + &
& scale_factor_ghcn_adj*anomaly_ghcn_adj
WRITE(UNIT = *, FMT = '(A)') 'Done.'
!%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
WRITE(UNIT = *, FMT = '(A)', ADVANCE = 'NO') 'Calculating area weights ... '
DO j = 1, nlat

```

```

DO i = 1, nlon
  area_weights(i,j) = (sin(deg2rad_r8b*lat_bnds(1,j)) - sin(deg2rad_r8b*lat_bnds(2,j)))* &
& (lon_bnds(2,i) - lon_bnds(1,i))
ENDDO
ENDDO

area_weights = deg2rad_r8b*abs(area_weights)/(4.0_r8b*pi_r8b)

WRITE(UNIT = *, FMT = '(A)') 'Done.'

!*****

WRITE(UNIT = *, FMT = '(A)', ADVANCE = 'NO') 'Loading land mask ... '

CALL check_nc_op(nf90_open(land_mask_file, nf90_nowrite, ncid))

CALL check_nc_op(nf90_inq_varid(ncid, 'lon', lon_varid))
CALL check_nc_op(nf90_inq_varid(ncid, 'lat', lat_varid))
CALL check_nc_op(nf90_inq_varid(ncid, 'land_fraction', land_mask_varid))

CALL check_nc_op(nf90_get_var(ncid, lon_varid, lon_mask))
CALL check_nc_op(nf90_get_var(ncid, lat_varid, lat_mask))
CALL check_nc_op(nf90_get_var(ncid, land_mask_varid, land_mask))

CALL check_nc_op(nf90_close(ncid))

WRITE(UNIT = *, FMT = '(A)') 'Done.'

!*****

WRITE(UNIT = *, FMT = '(A)', ADVANCE = 'NO') 'Interpolating land mask ... '

lon_mask_indices = (/ (i,i=1,nlon_land_mask)/)
lat_mask_indices = (/ (i,i=1,nlat_land_mask)/)
land_mask_interp = 0.0_r4b

DO j = 1, nlat
  DO i = 1, nlon

    lon_match = lon_mask > lon_bnds(1,i) .AND. lon_mask < lon_bnds(2,i)
    lat_match = lat_mask > lat_bnds(1,j) .AND. lat_mask < lat_bnds(2,j)

    nlon_match = COUNT(lon_match)
    nlat_match = COUNT(lat_match)

    IF(nlon_match > 0 .AND. nlat_match > 0) THEN
      land_mask_interp(i,j) = SUM(land_mask(PACK(lon_mask_indices, MASK = lon_match), &
& PACK(lat_mask_indices, MASK = lat_match))) &
& / (nlon_match*nlat_match)
    ENDIF

  ENDDO
ENDDO

WRITE(UNIT = *, FMT = '(A)') 'Done.'

!*****

WRITE(UNIT = *, FMT = '(A)', ADVANCE = 'NO') 'Scaling area weights ... '

max_land_area = sum(land_mask_interp*area_weights)
area_weights = area_weights*land_mask_interp/max_land_area

WRITE(UNIT = *, FMT = '(A)') 'Done.'

!*****

WRITE(UNIT = *, FMT = '(A)', ADVANCE = 'NO') 'Calculating spatial averages ... '

spatial_avg = missing
spatial_avg_common = missing

DO i = 1, ntime

  mask_ghcn = ABS(anomaly_ghcn(:, :, i) -missing_ghcn) > eps_r4b
  mask_ghcn_adj = ABS(anomaly_ghcn_adj(:, :, i) -missing_ghcn_adj) > eps_r4b
  mask = mask_ghcn .AND. mask_ghcn_adj

  area_common(i) = sum(area_weights, MASK = mask)
  area(i,1) = sum(area_weights, MASK = mask_ghcn)
  area(i,2) = sum(area_weights, MASK = mask_ghcn_adj)

  IF(area_common(i) > 0.) THEN

    spatial_avg_common(i,1) = SUM(anomaly_ghcn(:, :, i)*area_weights, MASK = mask)/area_common(i)
    spatial_avg_common(i,2) = SUM(anomaly_ghcn_adj(:, :, i)*area_weights, MASK = mask)/area_common(i)

  ENDIF

  IF(area(i,1) > 0) THEN
    spatial_avg(i,1) = SUM(anomaly_ghcn(:, :, i)*area_weights, MASK = mask_ghcn)/area(i,1)
  ENDIF

  IF(area(i,2) > 0) THEN
    spatial_avg(i,2) = SUM(anomaly_ghcn_adj(:, :, i)*area_weights, MASK = mask_ghcn_adj)/area(i,2)
  ENDIF

ENDDO

WRITE(UNIT = *, FMT = '(A)') 'Done.'

```

```

!&*****
!&*****
OPEN(UNIT = 10, FILE = 'ghcn_obs_avgs_common.dat')
OPEN(UNIT = 20, FILE = 'ghcn_obs_avgs_unique.dat')
OPEN(UNIT = 30, FILE = 'ghcn_obs_area_common.dat')
OPEN(UNIT = 40, FILE = 'ghcn_obs_area_unique.dat')

WRITE(UNIT = 10, FMT = '(A)') 'DATA: GLOBAL AVERAGE TEMPERATURE ANOMALIES'
WRITE(UNIT = 10, FMT = '(A)') 'SOURCES: GHCN, GHCN ADJ'
WRITE(UNIT = 10, FMT = '(A)') 'MASK: COMMON COVERAGE'
WRITE(UNIT = 10, FMT = '(A)')
WRITE(UNIT = 10, FMT = '(A4,1X,A5,1X,2(A10,1X))') 'YEAR', 'MONTH', 'GHCN', 'GHCN_ADJ'

WRITE(UNIT = 20, FMT = '(A)') 'DATA: GLOBAL AVERAGE TEMPERATURE ANOMALIES'
WRITE(UNIT = 20, FMT = '(A)') 'SOURCES: GHCN, GHCN ADJ'
WRITE(UNIT = 20, FMT = '(A)') 'MASK: UNIQUE COVERAGE'
WRITE(UNIT = 20, FMT = '(A)')
WRITE(UNIT = 20, FMT = '(A4,1X,A5,1X,2(A10,1X))') 'YEAR', 'MONTH', 'GHCN', 'GHCN_ADJ'

WRITE(UNIT = 30, FMT = '(A)') 'DATA: GLOBAL LAND SURFACE AREA BASED ON NON-MISSING TEMPERATURE ANOMALIES'
WRITE(UNIT = 30, FMT = '(A)') 'SOURCES: GHCN, GHCN ADJ'
WRITE(UNIT = 30, FMT = '(A)') 'MASK: COMMON COVERAGE'
WRITE(UNIT = 30, FMT = '(A)') 'UNITS: PERCENT OF EARTH'S LAND SURFACE'
WRITE(UNIT = 30, FMT = '(A)')
WRITE(UNIT = 30, FMT = '(A4,1X,A5,1X,A10)') 'YEAR', 'MONTH', 'COMMON'

WRITE(UNIT = 40, FMT = '(A)') 'DATA: GLOBAL LAND SURFACE AREA BASED ON NON-MISSING TEMPERATURE ANOMALIES'
WRITE(UNIT = 40, FMT = '(A)') 'SOURCES: GHCN, GHCN ADJ'
WRITE(UNIT = 40, FMT = '(A)') 'MASK: UNIQUE COVERAGE'
WRITE(UNIT = 40, FMT = '(A)') 'UNITS: PERCENT OF EARTH'S LAND SURFACE'
WRITE(UNIT = 40, FMT = '(A)')
WRITE(UNIT = 40, FMT = '(A4,1X,A5,1X,2(A10,1X))') 'YEAR', 'MONTH', 'GHCN', 'GHCN_ADJ'

k = 1
DO i = year0, yearf
  DO j = 1, 12
    WRITE(UNIT = 10, FMT = '(I4,1X,I5,1X,2(F10.5,1X))') i, j, spatial_avg_common(k,:)
    WRITE(UNIT = 20, FMT = '(I4,1X,I5,1X,2(F10.5,1X))') i, j, spatial_avg(k,:)
    WRITE(UNIT = 30, FMT = '(I4,1X,I5,1X,F10.4)') i, j, area_common(k)
    WRITE(UNIT = 40, FMT = '(I4,1X,I5,1X,2(F10.4,1X))') i, j, area(k,:)
    k = k + 1
  ENDDO
ENDDO

CLOSE(UNIT = 1)
CLOSE(UNIT = 2)

!&*****
END PROGRAM mask_ghcn_obs

```