Evaluating Temperature Forecast Accuracy in WXSIM

Error Statistics Primer

There are several ways of expressing temperature forecast errors. The simplest is net error, or just the predicted value minus the actual one, so that positive values mean the forecast was too warm, and negative ones too cold. Another is the absolute error, simply the absolute value (how far from zero, as an always-positive number) of the net error. If the forecast is for a high of 31, and the actual value is 34, the net error is -3 and the absolute error is just 3.

Once we are considering more than one day, we may want to find some type of average. The mean net error (MNE) is just the arithmetic mean of the net errors. So, let's say over a period of 5 days, the forecasts were 31, 37, 33, 27, and 32, and the corresponding actual values were 34, 39, 32, 29, and 28. The mean net error would be (-3-2+1-2+4)/5 = -2/5 = -0.4. The mean absolute error (MAE) is the mean of the absolute errors, or (3+2+1+2+4)/5 = 12/5 = 2.4.

Another type of mean error is root-mean-square (RMS). This weights large errors more by squaring them, and is found by taking the square root of the average of the squares of the errors. In the example above, we get (9+4+1+4+16)/5 = 34/5 = 6.8, and then take the square root to get 2.6077. Note that this is generally larger than the mean absolute error, usually by about 25% in real weather data, though in this sample it was only by about 8%. It is obviously important to know whether we are discussing MAE or RMS.

An interesting debate is which of these is better. RMS may be considered a "tougher" test because the number is bigger, but that's of course a relative thing. RMS could be considered superior if it is regarded as four times as bad to be twice as far off. This is close to the idea of having some threshold, where everything under it is "OK", but beyond it is not. However, many if not most applications of temperature data (like heating or cooling degree days, output of power plants, etc.) show fairly direct proportions between the amount of error and economic consequences, so there is a good argument to be made that MAE is superior to RMS (that's my opinion); certainly MAE is easier to compute!

Closely related to RMS, and quite important, is standard deviation of error (STDEV). The simplest version of this is the "uncorrected sample standard deviation", calculated by finding the square root of the average of the squares of the differences of the values from the mean. In our example above, we first find the mean of the net errors, which was just MNE, = 0.4 (also called the "bias"). We then subtract this from the individual errors, getting 2.6, 1.6, -1.4, 1.6, and -4.4. Square all these and add them up to get 33.2 and divide by 5 to get 6.64. Finally, take the square root to get 2.5768.

Actually, there are other definitions of standard deviation, such as the "population" one, where we divide the sum of the squares of the differences by one less than the number of them (4, in this case). This would give 2.881 in our example. There are also ways to "unbias" the calculation. At this point, we are "beyond my pay grade", statistically speaking, and I do not have a clear idea of which of these is most valid for the current application (see the Wikipedia article on standard deviation, or ask – or be! – a statistician). However, the beauty of this is that, as the number of values gets larger, the differences among these various versions shrinks, becoming really tiny once we have hundreds of values, so I will not worry more about it here. Instead, I'll stick with the first one, because it has a neat property which simplifies calculations … this STDEV is just the square root of the difference of the squares of RMS and MNE! Try it: (2.6077^2 – 0.4^2)^0.5 = 2.5768.

A good thing about STDEV of error is that it tells you what the RMS error would be if you completely eliminated the bias. Usually, in forecasting, the bias is something we can adjust, but all the factors that cause variations in the error are hard to control. In the example above, the seemingly substantial improvement in net error we would get by correcting the bias of -0.4 would yield an improvement of only 2.6077-2.5768 = 0.0309 in RMS, and slightly less than that in MAE. On the other hand, consider a RMS of 3 with a bias of 2. Correcting that bias would bring the RMS down from 3 to the square root of 5, or 2.24. Another example is that correcting a bias half the size of the RMS reduces RMS and MAE to less than 87% of their former values.

So, what are good values of these measures of forecast accuracy? The answer varies dramatically by location and season. In a monotonous tropical climate, the standard deviation of the temperature (say, daily maximum) itself may be less than 2 Celsius degrees C (3.6 Fahrenheit degrees), so that merely guessing that it will be the long term average ("climatology") or the same as yesterday ("persistence") is already that good of a forecast, in spite of showing no "skill". On the other hand, in a mid- or high latitude location in winter, such an RMS error could indicate considerable skill, as day-to-day variations are several times that. A good measure of skill is 1 – RMSE(using our forecast)/RMSE(using climatology). RMS using climatology is really just STDEV of the day to day variations. So, if we manage an RMS error of 2 degrees, when standard deviation of day-today temperature in that location at that time of year is 5 degrees, our skill score would be 1-2/5 = 0.6. In another way of looking at it, we might also say the forecast is 5/2 = 2.5 "times better" than climatology. Of course, if all we want is to have the temperature forecast within certain limits, the "skill" may be of little practical interest, outside the world of forecasting connoisseurs!

To answer, then, the question of what's a good value (after the above digression), temperature forecasts for mid latitude sites, from government weather offices or from skilled private forecasters may have, for the first 24 hours, RMSE's of less than 1.5 C (2.7 F) much of the year. Continental locations in winter are likely to have larger errors, and errors of course grow with forecast lead time, so that by 5 days out, RMSE's may be close to 3 C (5.4 F) and may show skill under 0.5. Recall that MAE is likely to be only about 80% of RMSE.

To conclude this section, we mention a very simple and intuitive measure of error (or skill): the percentage of forecasts with errors less than a specified threshold. For example, we might say that some source got the maximum temperature within 2 degrees of actual 70% of the time, while another (less skillful source) accomplished this only 50% of the time. This is elegant and understandable, but doesn't convey all the information we might like (such as bias).

The Origins of Error in WXSIM

I first started writing WXSIM (in the early 1980's) not really as a forecasting tool, but rather as an experiment in applying physics to simulate the diurnal temperature curve. In this original form, "weather conditions" (i.e. cloud cover, wind) were *input* for the program, not output! Even as I started to consider it a temperature forecasting tool, I was still putting weather conditions into the program. My first mention of it publicly, on an AOL weather forum in 1994, was ridiculed by someone (a meteorologist or meteorology student, I think) for that very reason. He didn't understand what I was trying to do, and never responded to my explanation. Thankfully, a meteorologist (Mark Ewens) with the NWS did appreciate what I was doing, as he had attempted to do something similar himself. He was helpful and encouraging to my early efforts to present it as a sort of forecasting tool.

I worked obsessively for years identifying and incorporating variables which could affect the diurnal temperature curve. This included establishing "tune-able parameters" which could customize the model's output to fit particular sites. Getting the temperatures right was my main goal, but other things show diurnal variations, too. These include dew point (and secondarily relative humidity), wind speed, and some types of cloud cover (i.e. fair weather cumulus and nighttime and morning stratus). Import of air (advection) with different temperature and dew point was another vital piece of the puzzle, and I created a rather sophisticated routine to handle this.

I had a huge amount of data to work with. Some of it was my own, but most was a set of CD-ROM's from the national Climatic Data Center. I wrote software to search these data by criteria and do averages, to create a wide variety of "model days" for various locations under a wide range of conditions. After a great deal of tweaking, I got WXSIM to fit almost all of these curves to within a degree or two Fahrenheit (generally less than a degree Celsius) at all hours of the day.

So, *given* correct cloud cover, wind speed, upper air temperatures, advection conditions, soil moisture (via recent rainfall amounts), etc., WXSIM could now "forecast" (simulate, really) daily max and min temperatures with errors generally less than one degree Fahrenheit for most of these scenarios. (Most of this work was many years ago, and I don't have exact figures, but this is the kind of accuracy I was getting).

The problem for using it for forecasting now became … "so just what will the cloud, wind, etc. conditions actually be?" Other than some fairly primitive routines mentioned earlier, these types of "weather" were hardly forecast by WXSIM; so, when I used it to forecast, I put them in (by typing them in as input or entering changes with keystrokes). I would consult other forecasts and sometimes satellite or radar images to get these data into the program. As the internet became available, I was able to get much of this information in the form of model output statistics (MOS) from various U.S. numerical models. I became skilled at this, and was able to generate temperature forecasts that were generally better than those in the MOS I was using, and often (especially in the short term) better than the National Weather Service humans were producing. With the right input, WXSIM was proving itself to be an excellent temperature forecasting tool. Other temperature-related things, like accumulation and melting of snow, cooling effect of precipitation, formation of dew and frost, etc. were modeled successfully as well.

I started selling WXSIM on a small scale in 1994, with a pick-up in 2000 as Tim Vasquez of weathergraphics.com discovered it and advertised it on his site. Users at this time understood the need to enter their own data. A convenient innovation about that time was the Interrupt Planner, where these changes could be clicked onto a graph, allowing the program to run without manual interruption. Then, I developed routines to parse FOUS data blocks and READY meteogram data for models such as the GFS (called "AVN" at the time) and NAM (called "ETA" at the time).

This meant the user was now a bit removed from the forecast process, which was a philosophical change in something I had been describing as an "interactive local atmospheric model". It also meant that all those "external" variables (cloud cover, wind direction, etc.) were in the hands of the chosen model data. This meant that errors in that data translated readily into errors in WXSIM's temperature output. Of course, there's no point in a "blame game" regarding forecasts busts being due to bad imported forecast data, because the information has to come from somewhere. What *is* important to realize here, though, is that error in WXSIM can be meaningfully grouped into two types: native model error (usually very small) and external model error (usually fairly small, but rather variable).

Several years ago, I took another very important step, by arranging automated import of data and running of the program, freeing the user from almost all interaction. This is very convenient and understandably increased the program's popularity. A big part of this was the created of the site-specific GFS data, with world-wide coverage, thanks to Chris McMahon of weather-watch.com. A minor drawback of this was that many users either forgot or never learned how to grab MAPS or NAM (via READY) data that had been helpful. The loss in accuracy due to this was quite small, and only applies to North American users, but it illustrates how convenience can result in a trade-off.

Much of my work the last several years has consisted of honing data gathering routines, adding new routines to handle off geographical situations, fulfilling user requests for types of output, and adding a "learning routine" aimed at reducing the various biases which an imperfect customization leaves behind. I've barely touched the central, native model in many years, as roughly two decades of work had gone into "perfecting" it.

Keeping Track of Accuracy

First, one must decide upon and clearly define what, exactly to evaluate. The most common item (for WXSIM and other providers) is daily maximum and minimum temperature. These may seem to be straightforward enough, but in fact there are multiple definitions of these. For U.S. (and most countries) record-keeping purposes, these are defined as the highest and lowest temperatures during the 24 hour period from midnight to midnight (local standard time). Australia and possibly some other countries, for historical reasons, use 9 AM to 9 AM. This may have been for the observer's convenience, but it actually introduces some problems, especially in winter, when 9 AM is not sufficiently long after sunrise to ensure (more or less) that two separate days will not "share" minimum temperatures. It produces a cold-bias (I don't know how much) on average daily minimum temperature.

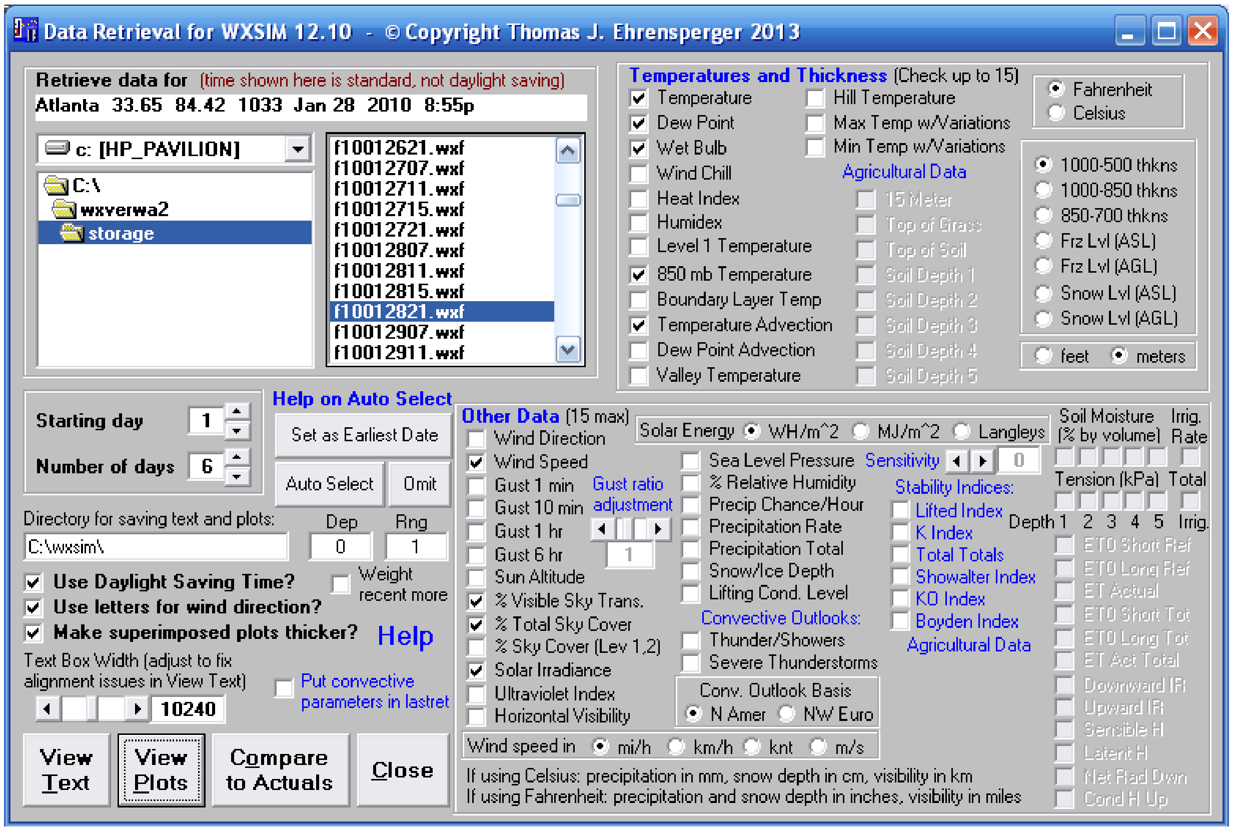
However, neither of these is typically used for forecast purposes. Instead, most commonly, the minimum temperature is an "overnight" value, for a period beginning at perhaps 7 PM and ending at perhaps 8 AM (that's the definition adopted for U.S. MOS products, and basically for the public forecasts as well … note 8 AM is to allow for sunrise to have passed, in most cases). The maximum is usually a 7 AM to 7 PM maximum. These hours may be a bit different for different locations, and/or may be only vaguely defined, but are in fact pretty much what the public expects, given the timing of the diurnal cycle (minimum just after sunrise, maximum in mid-afternoon).

WXSIM provides both 24 hour max and min, and “AM lows and PM highs” (using exactly the definition for MOS, described above). The popular plaintext.txt output uses the latter. However, WXSIM’s self-evaluation tool goes with the convenience of the midnight-to-midnight version, for many programming-related reasons. Anyone studying accuracy of WXSIM or any other source should be careful to keep these ideas in mind.

I have, at various times in the past, laboriously and faithfully recorded the WXSIM’s forecasts (for the official site at Atlanta Hartsfield-Jackson Airport), along with a few other sources’ forecasts and the actual resulting temperatures (using the AM/PM definitions). Using spreadsheets, I’ve done a lot of analysis and also nicely cataloged WXSIM’s strengths (and weaknesses).

The Data Retrieval Program (wret.exe)

From very early on in WXSIM’s development, I had been writing a separate program for more detailed and versatile display of WXSIM’s output, as saved in text files with the extension .wxf. There are now over 80 different variables, all saved from previous forecasts, which can be selected for viewing either as text or graphically in wret.exe. Additionally, by right-clicking on the plotted graphs, you can pull up vertical soundings (either rectangular or “skew-T”) for every half-hour of the forecast. You can even superimpose graphs on each other for direct comparisons of multiple forecasts (each shown with a different thickness line), which may be helpful for seeing the effects of changes to setting in a given forecast. Here is the main form of wret.exe:


Data Retrieval for WXSIM 12.10 - © Copyright Thomas J. Ehrensperger 2013

Retrieve data for (time shown here is standard, not daylight saving)
Atlanta 33.65 84.42 1033 Jan 28 2010 8:55p

c: [HP_PAVILION]
C:\
wxverwa2
storage

f10012621.wxf
f10012707.wxf
f10012711.wxf
f10012715.wxf
f10012721.wxf
f10012807.wxf
f10012811.wxf
f10012815.wxf
f10012821.wxf
f10012907.wxf
f10012911.wxf

Temperatures and Thickness (Check up to 15)
- [✓] Temperature
- [✓] Dew Point
- [✓] Wet Bulb
- [ ] Wind Chill
- [ ] Heat Index
- [ ] Humidex
- [ ] Level 1 Temperature
- [✓] 850 mb Temperature
- [ ] Boundary Layer Temp
- [✓] Temperature Advection
- [ ] Dew Point Advection
- [ ] Valley Temperature
- [ ] Hill Temperature
- [ ] Max Temp w/Variations
- [ ] Min Temp w/Variations

Agricultural Data
- [ ] 15 Meter
- [ ] Top of Grass
- [ ] Top of Soil
- [ ] Soil Depth 1
- [ ] Soil Depth 2
- [ ] Soil Depth 3
- [ ] Soil Depth 4
- [ ] Soil Depth 5

[✓ Fahrenheit] Celsius

[✓ 1000-500 thkns] 1000-850 thkns 850-700 thkns Frz Lvl (ASL) Frz Lvl (AGL) Snow Lvl (ASL) Snow Lvl (AGL)

feet [✓ meters]

Help on Auto Select

Starting day 1
Number of days 6

Set as Earliest Date
Auto Select
Omit

Directory for saving text and plots: C:\wxsim\
Dep 0
Rng 1

- [✓] Use Daylight Saving Time?
- [✓] Use letters for wind direction?
- [✓] Make superimposed plots thicker?
- [ ] Weight recent more

Help

Text Box Width (adjust to fix alignment issues in View Text) 10240

- [ ] Put convective parameters in lastret

View Text
View Plots
Compare to Actuals
Close

Other Data (15 max)
- [ ] Wind Direction
- [✓] Wind Speed
- [ ] Gust 1 min
- [ ] Gust 10 min
- [ ] Gust 1 hr
- [ ] Gust 6 hr
- [ ] Sun Altitude
- [✓] % Visible Sky Trans.
- [✓] % Total Sky Cover
- [ ] % Sky Cover (Lev 1,2)
- [✓] Solar Irradiance
- [ ] Ultraviolet Index
- [ ] Horizontal Visibility

Gust ratio adjustment 1

Solar Energy [✓ WH/m^2] MJ/m^2 Langleys

- [ ] Sea Level Pressure
- [ ] % Relative Humidity
- [ ] Precip Chance/Hour
- [ ] Precipitation Rate
- [ ] Precipitation Total
- [ ] Snow/Ice Depth
- [ ] Lifting Cond. Level

Sensitivity 0

Stability Indices:
- [ ] Lifted Index
- [ ] K Index
- [ ] Total Totals
- [ ] Showalter Index
- [ ] KO Index
- [ ] Boyden Index

Convective Outlooks:
- [ ] Thunder/Showers
- [ ] Severe Thunderstorms

Agricultural Data

Conv. Outlook Basis [✓ N Amer] NW Euro

Wind speed in [✓ mi/h] km/h knt m/s

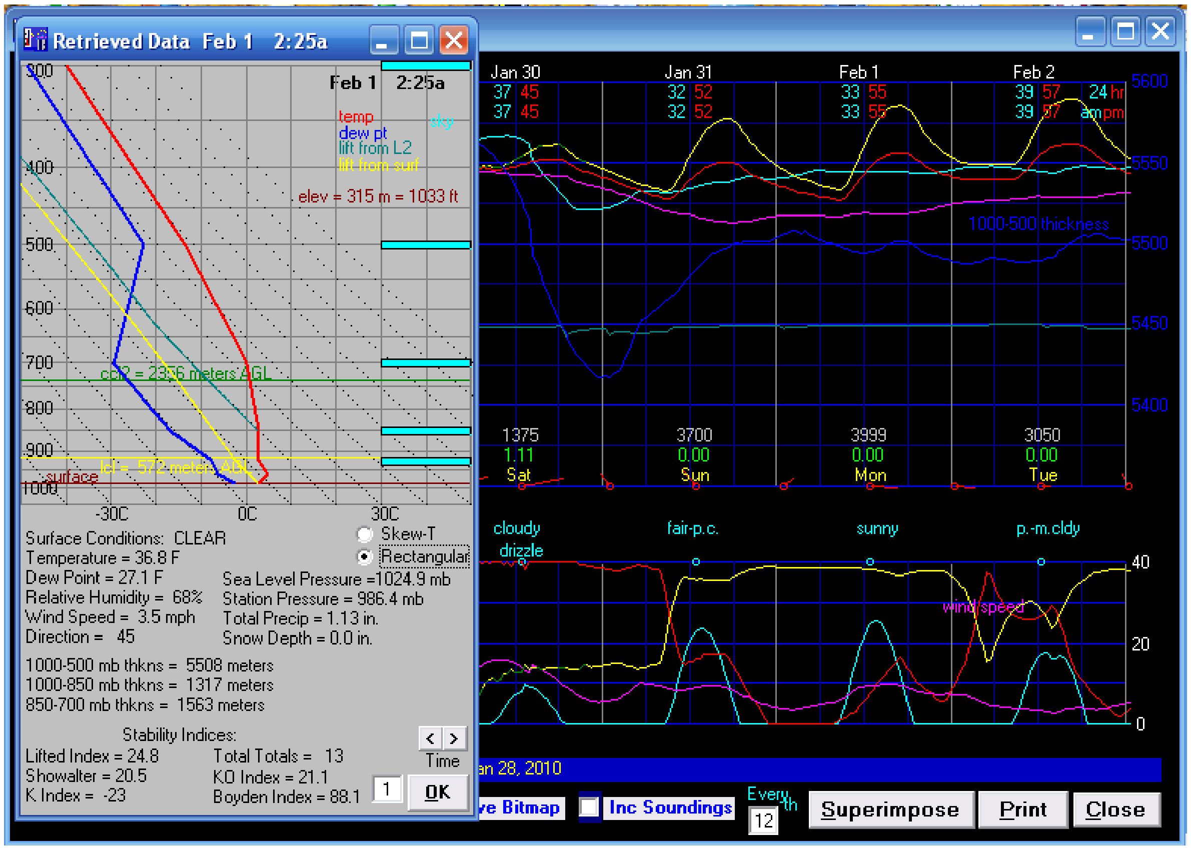
Soil Moisture (% by volume) Irrig. Rate
Tension (kPa) Total
Depth 1 2 3 4 5 Irrig.

- [ ] ET0 Short Ref
- [ ] ET0 Long Ref
- [ ] ET Actual
- [ ] ET0 Short Tot
- [ ] ET0 Long Tot
- [ ] ET Act Total
- [ ] Downward IR
- [ ] Upward IR
- [ ] Sensible H
- [ ] Latent H
- [ ] Net Rad Dwn
- [ ] Cond H Up

If using Celsius: precipitation in mm, snow depth in cm, visibility in km
If using Fahrenheit: precipitation and snow depth in inches, visibility in miles

As in WXSIM, instant help is available by clicking on any blue captions. Some items (soil and agricultural) are greyed out here because the forecast about to be viewed in from 2010, before I created the modules for these, and such data simply isn’t in the forecast. WXSIM’s professional mode allows use of the soil and agricultural features, so resulting forecasts can be fully displayed in wret.exe. You can view text or plots, and in fact when wret.exe is initiated by WXSIM through the ‘Graphic’ option (right after completion of a forecast, with check box for that on WXSIM’s Output form), the

View Text and View Plots routines are executed briefly to produce the file lastret.txt (used by many third-party PHP scripts and a graphic called lastret.bmp, which generally looks like this:



Retrieved Data Feb 1 2:25a

Surface Conditions: CLEAR
Temperature = 36.8 F
Dew Point = 27.1 F
Relative Humidity = 68%
Wind Speed = 3.5 mph
Direction = 45

Sea Level Pressure =1024.9 mb
Station Pressure = 986.4 mb
Total Precip = 1.13 in.
Snow Depth = 0.0 in.

1000-500 mb thkns = 5508 meters
1000-850 mb thkns = 1317 meters
850-700 mb thkns = 1563 meters

Stability Indices:
Lifted Index = 24.8
Showalter = 20.5
K Index = -23
Total Totals = 13
KO Index = 21.1
Boyden Index = 88.1

Skew-T [✓ Rectangular]

In this case, the mouse was right-clicked under February 1, to pull up the vertical sounding superimposed on the left side of the graph. Wret.exe can generate a whole series of soundings if ‘Inc Soundings’ is checked.

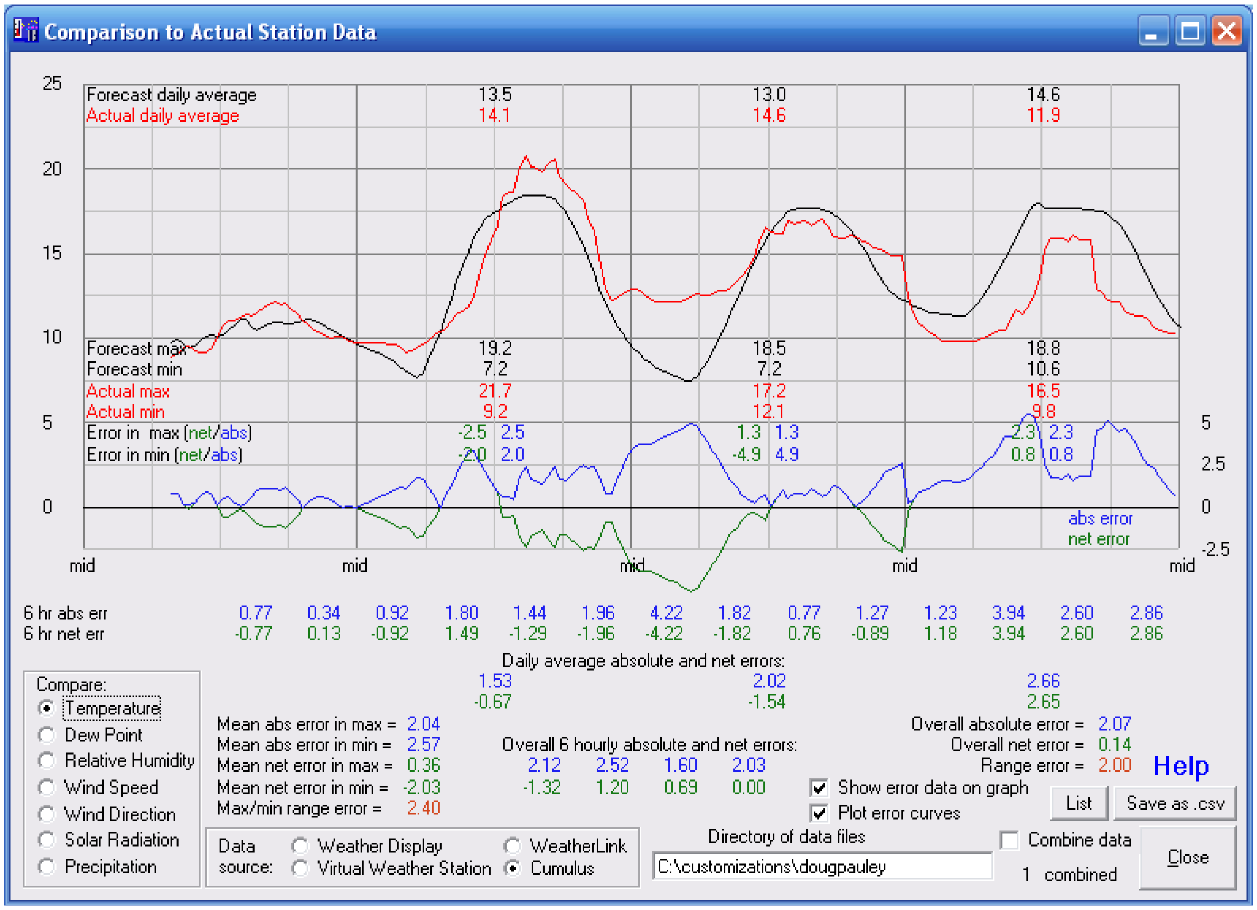
Comparisons to Actual Data (from your home weather station)

As many users began to have home weather stations and software for logging the data, I decided to also develop a way to display forecasts compared directly against the “verification data” (what actually happened). This can now be done with logged data from WeatherLink, Weather Display, Virtual Weather Station, or Cumulus. I chose seven variables for display and analysis: temperature, dew point, relative humidity, wind speed, wind direction, solar radiation (an indirect measure of daytime cloud cover), and precipitation.

To use this feature, first use the tools (drive, directory, and file list boxes) to browse to find a forecast you want to view, and click on it. Also, select the number of days you wish to view, which should not be more than the length of the forecast and should not take you up past today, as there must be actual data against which to compare the forecast! Having done this, click the Compare to Actuals button (see the form shown on page 4, above, for illustration). A form should appear, and it may be blank the first time you try this, because you will need to specify the software type and the path to its log files. A partial exception is WeatherLink, as the required files will already be in the current directory, at least if you have recently (preferably same day) run WXSIMATE to grab data from WeatherLink. Once you have made your choices, click Close to exit the form, and simply try again. As long as you keep using the same software (and assuming you exit wret.exe properly when you are through), these settings will be saved.

Below is an example of such a comparison. The black curve is the forecast, and the red one (rather jagged, probably due to variations in cloud cover) is the actual data. Other variables can be selected for display using the “radio” buttons at lower left. The various figures shown are color-coded for easy identification, and were essentially explained earlier in this article.

The particular forecast shown is not a terribly good one, but there is a ready “excuse”, at least for the defense of WXSIM’s native routines: cloud cover was not forecast very well, especially after about 10 PM of the second day. The black curve continues to drop, but the red, real data levels off suddenly and stays that way all night. This is almost certainly the result of an unexpected (that is, by the GFS data on which WXSIM was heavily relying on for cloud cover information) overcast appearing at that time. WXSIM basically uses this GFS cloud cover - tweaked a bit using GFS layer relative humidities, and with user-specifiable bias adjustments – for its cloud forecast, and the internally modeled temperatures respond accordingly. WXSIM’s only native cloud cover routines are the auto cumulus and auto stratus, which are diurnally driven and unaware of large-scale weather patterns (in fact I rarely recommend using them, except in monotonous weather patterns). It is also interesting to note the apparent sudden clearing of the sky just before midnight on day 3. Note also that data for day 1 is limited (i.e., max and min are not given) by the fact it’s not a whole 24 hour period.



Forecast daily average	13.5	13.0	14.6
Actual daily average	14.1	14.6	11.9
Forecast max	19.2	18.5	18.8
Forecast min	7.2	7.2	10.6
Actual max	21.7	17.2	16.5
Actual min	9.2	12.1	9.8
Error in max (net/abs)	-2.5 2.5	1.3 1.3	2.3 2.3
Error in min (net/abs)	-2.0 2.0	-4.9 4.9	0.8 0.8

6 hr abs err	0.77	0.34	0.92	1.80	1.44	1.96	4.22	1.82	0.77	1.27	1.23	3.94	2.60	2.86
6 hr net err	-0.77	0.13	-0.92	1.49	-1.29	-1.96	-4.22	-1.82	0.76	-0.89	1.18	3.94	2.60	2.86

Daily average absolute and net errors:

1.53	2.02	2.66
-0.67	-1.54	2.65

Compare:
- [✓ Temperature]
- Dew Point
- Relative Humidity
- Wind Speed
- Wind Direction
- Solar Radiation
- Precipitation

Mean abs error in max = 2.04
Mean abs error in min = 2.57
Mean net error in max = 0.36
Mean net error in min = -2.03
Max/min range error = 2.40

Overall 6 hourly absolute and net errors:

2.12	2.52	1.60	2.03
-1.32	1.20	0.69	0.00

Overall absolute error = 2.07
Overall net error = 0.14
Range error = 2.00

Help

[✓ Show error data on graph]
[✓ Plot error curves]

List | Save as .csv

Data source: Weather Display | WeatherLink | Virtual Weather Station | [✓ Cumulus]

Directory of data files: C:\customizations\dougpauley

Combine data
1 combined

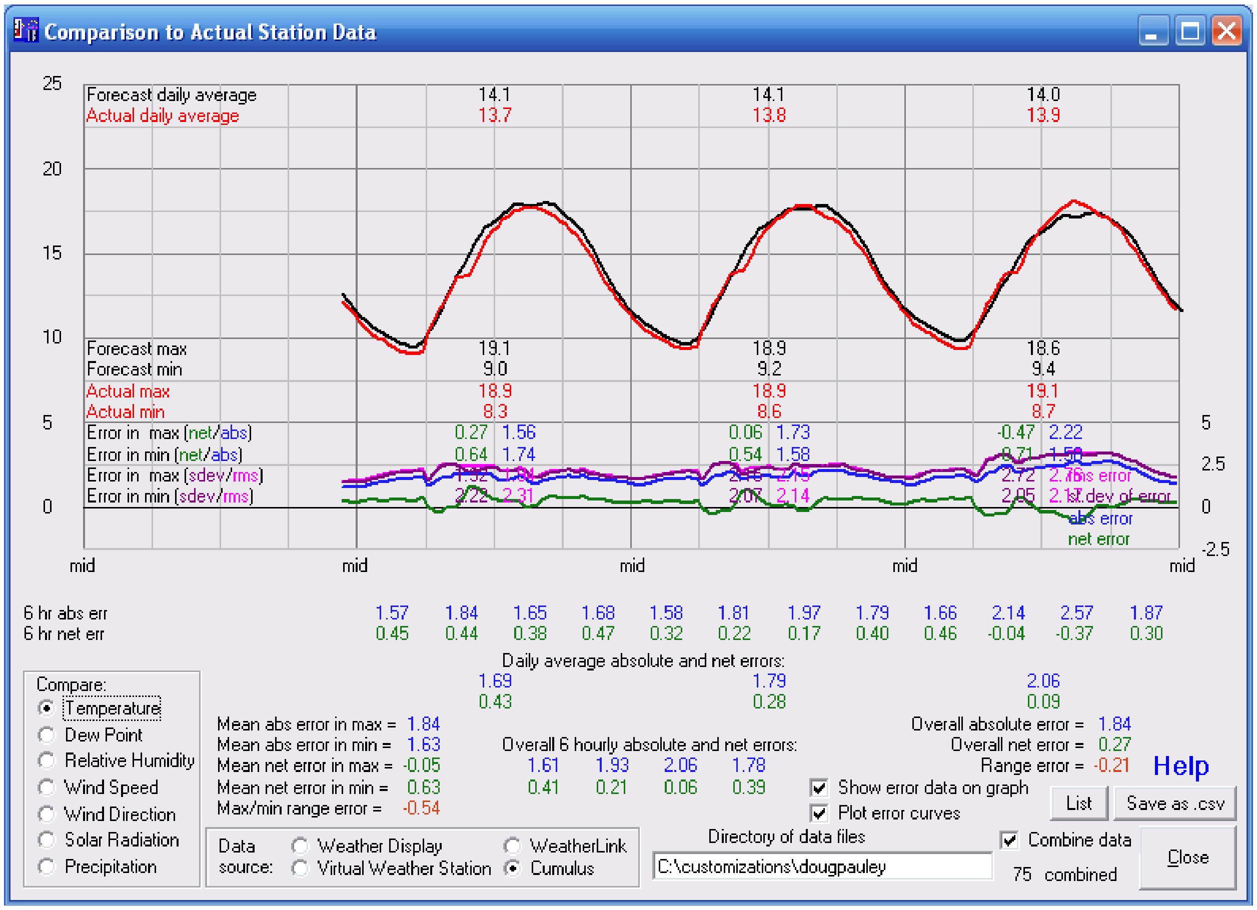
Close

Because so much of the inaccuracy in this forecast is erratic and “beyond WXSIM’s control”, so to speak, it tells very little about how well WXSIM is customized for this location (Wetherby, England). Much more data is needed. In order to meet this requirement, we need to combine a lrge number of forecasts; so, check the “Combine data” box! Then click Close, and select another forecast. You can do this over and over again.

Below is the result of doing this 75 times – 2 or 3 forecasts per day for a whole month (May 19 to June 19, 2013). Now the black curve is an excellent match to the red one, and the accuracy of WXSIM’s internal routines is plain to see. The forecast maximum temperatures (which are properly a bit higher than the curves actually show, to take into account short-term random fluctuations) average just 0.05 degrees C too cool – virtually perfect. The forecast minimum temperatures are just a bit too warm, by 0.63 degrees. This could be regarded as a very good customization, but still marginally worth making another pass at, to tweak those minimum temperatures.

A wealth of statistical information is visible (and you can use check boxes to show less at once if it's too cluttered), including all the error statistics described at the beginning of this article. While the net errors (green) are tiny, the mean absolute ones (blue) are understandably larger, because of "random" (here meaning beyond our ability to control) fluctuations in things like cloud cover, timing of rain showers, wind shifts, etc. Still, these errors reach as much as 2 degrees only on day 4. By any reasonable standard, these were good forecasts.

Something interesting is visible in the red curve. Each day, just after 9 AM, there is a little "ledge" in the curve. I haven't asked the customer to figure this out, but my experience suggests it's a shadow falling on the instrument shelter during that time. All sorts of little thermometer placement issues can become apparent in these analyses!



Comparison to Actual Station Data

	Day 2	Day 3	Day 4
Forecast daily average	14.1	14.1	14.0
Actual daily average	13.7	13.8	13.9
Forecast max	19.1	18.9	18.6
Forecast min	9.0	9.2	9.4
Actual max	18.9	18.9	19.1
Actual min	8.3	8.6	8.7
Error in max (net/abs)	0.27 1.56	0.06 1.73	-0.47 2.22
Error in min (net/abs)	0.64 1.74	0.54 1.58	0.71 1.58
Error in max (sdev/rms)	1.92 1.84	2.16 2.15	2.72 2.76
Error in min (sdev/rms)	2.22 2.31	2.07 2.14	2.05 2.17

6 hr abs err	1.57	1.84	1.65	1.68	1.58	1.81	1.97	1.79	1.66	2.14	2.57	1.87
6 hr net err	0.45	0.44	0.38	0.47	0.32	0.22	0.17	0.40	0.46	-0.04	-0.37	0.30

Daily average absolute and net errors:

1.69	1.79	2.06
0.43	0.28	0.09

Compare:
[✓ Temperature]
Dew Point
Relative Humidity
Wind Speed
Wind Direction
Solar Radiation
Precipitation

Mean abs error in max = 1.84
Mean abs error in min = 1.63
Mean net error in max = -0.05
Mean net error in min = 0.63
Max/min range error = -0.54

Overall 6 hourly absolute and net errors:

1.61	1.93	2.06	1.78
0.41	0.21	0.06	0.39

Overall absolute error = 1.84
Overall net error = 0.27
Range error = -0.21

Help

☑ Show error data on graph
☑ Plot error curves

List | Save as .csv

Data source: Weather Display | Virtual Weather Station | WeatherLink | [✓ Cumulus]

Directory of data files
C:\customizations\dougpauley

☑ Combine data
75 combined

Close

When I do re-customizations (or when I want to diagnose a more serious forecast error), this is the kind of analysis I do, using large numbers of files sent to me. I then run scenarios before and after making tweaks to settings or to the cty.fdt file, to find ways to compensate for the systematic errors (biases) that show up in the analysis (which, by the way, usually includes doing the analysis separately for different seasons). I'm pleased to report that the good example above was on the first try! The customer did actually decide to have me fine-tune it after all, though; he's a perfectionist! One might ask just how much improvement one might see in tweaking an already-good customization like this. A good hint can be gotten by comparing the standard deviation and root-mean-square errors for a given item - like day 2 minimum temperature. In this case, the RMSE (generally about 25% larger than the MAE) is 2.31, with a standard deviation of error of 2.22. This suggests that a simple, successful bias correction could reduce the RMSE to 2.22. The MAE would likely shrink proportionally, from 1.74 to about 1.67, and improvement of just under 4 percent. That's not a lot, but small improvements are in fact hard to come by in weather forecasting, so some may find it worth the trouble. In re-customizing, I can usually get rid of most of the systematic error. Sometimes, I am able to identify other issues which may even decrease the standard deviation a bit, yielding slightly more improvement.

I have done some spreadsheet work to better define how much RMSE or MAE improvement should result from correction of bias (there's probably some theoretical approach one could use, but I don't have much training in statistics). I find that if the bias (mean net error) is as large as the standard deviation, an improvement of about 30% can be expected. If the bias is half the StDev, only about 7 or 8 percent improvement is likely. If the bias is only ¼ of the StDev, only about 2% improvement is possible (the improvement appears to be proportional to the square of the bias:StDev ratio). I think this depends on the distribution of data, though, as the above example shows more improvement than this pattern suggests. Look for the green heading on page 12 to see how to do this more easily!

The "Learning Routine"

It occurred to me (and I think some users suggested this, too) that this analysis could be used to generate "correction factors", to improve the forecasts *without* my doing an enhanced customization. This entails some risk to accuracy, as the user must instigate this correction, and there are many things that can go wrong, ranging from bad or missing log file data to forecasts made with little input data due to internet data source difficulties. However, I found ways to avoid most of these problems, and instituted the option, for WXSIM's professional mode only.

I will first describe shortly exactly how to use the learning routine, before explaining it in detail and discussing related issues, but first, brief explanations of a few options are in order. REfering to the image on page 4 above, note there is an 'Omit' button. Highlighting a forecast and using this button right before an Auto Select run tells the program to leave that forecast out of the analysis (it could be experimental, or for a different site, for example). 'Dep' and 'Rng' have a very specialized purpose and will probably rarely be set at anything other than the defaults of 0 and 1. However, I use it, because I am trying to forecast for the Atlanta airport, a few miles from the weather station on the roof of the school where I teach. The station is 50 feet (15 meters) above the ground, and I've documented that KATL runs a fraction of a degree cooler, and has about 8% more diurnal temperature range that this rooftop station does. So, I enter a small negative number for 'Dep' and about 1.08 for 'Rng', so that the data will be adjusted accordingly, and it will "learn" for the airport, not the school. Finally, there is a check box for 'weight recent more'. The idea here is that the possobo;ity of persistence of recent weather patterns or other temperature-affecting conditions would suggest using recent data, and yet older data is surely worth something (and there's much more of it!). If you check this box, data from the last 10 days will be weighted more than data from the previous 30 days, which in turn will be weighted more than anything before that. The weightings are somewhat complicated and will not described here. However, I've come to question the usefulness of this. In "theory" it's a good idea, but in practice, recent anomalous weather might "confuse" the learning routine. I suspect these weightings may be advantageous *if* cloud and wind forecasts have been reliable lately. I'm eager to find out more about what works best in practice. For the moment, though, I'm personally "playing it safe" by leaving the box unchecked, and using at least two months of forecasts to learn from. One other brief note: to use the learning routine, the time interval in the .wxf file needs to be 30 minutes. Most combinations of time interval and iterations per interval indeed yield this, but a few (like 1 hour and 3 iterations) can yield 20. So, make sure it's 30 minutes if you want to use this routine.

OK, back to just how to use the routine! Here are the steps:

(1) Select 4 as the number of days (other numbers are OK for other purposes, but not here!).

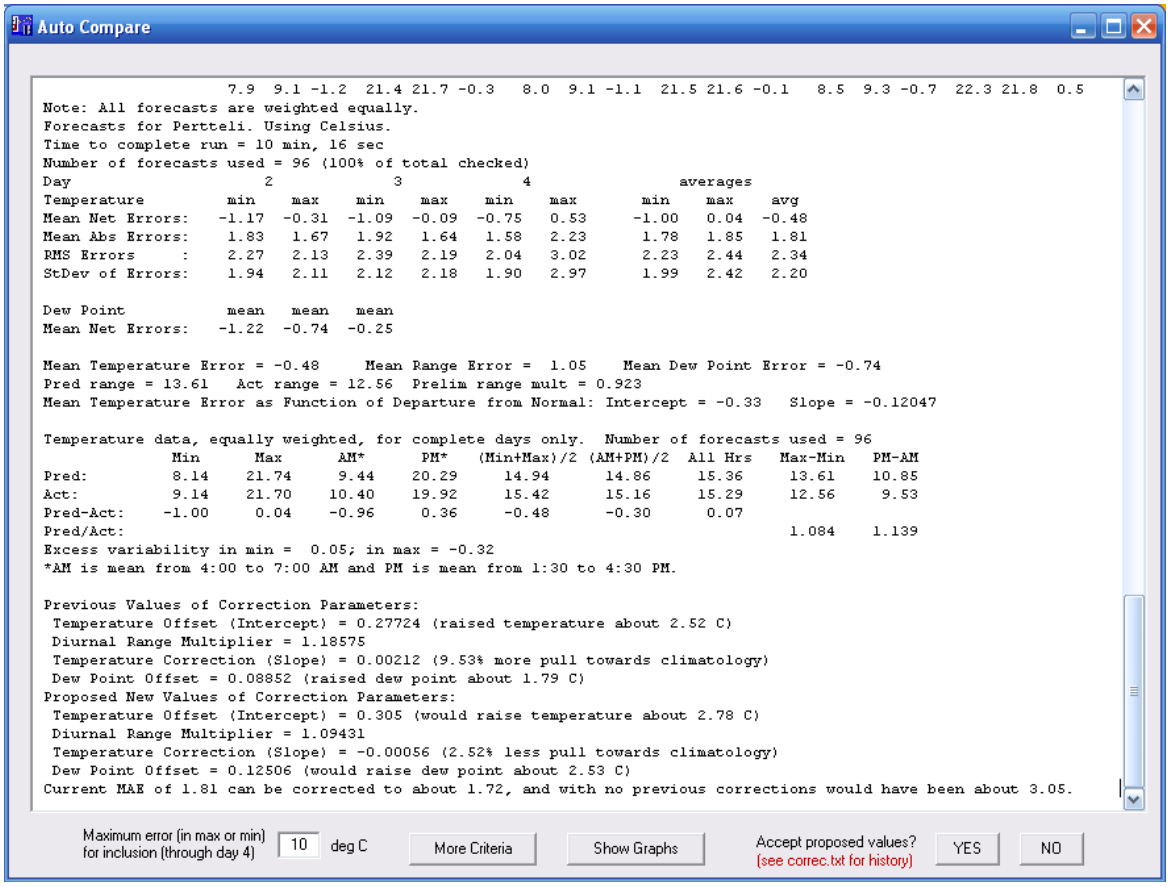
(2) Select a forecast from the list box and click Compare to Actuals. If changes in path or software-type information are needed, make them as described above, but take this step (of displaying one comparison) *even if* you've set the path, etc. in the past. One such display is enough to set it for the current session, though, even if you do multiple analyses. Do NOT check 'Combine data'.

(3) Find the first forecast you want considered (preferably from two of three months ago). Click on it in the list box and then click the "Set as Earliest Date" button.

(4) Count backwards from today's date, and click on a forecast made 4 days before today – meaning for example, the 21st of the month, if today is the 25th. Then click the "Auto Select" button.

You should now see the blue highlighting scrolling through the list box. This may take a while, perhaps even several minutes if there are a lot of forecasts in the folder. The program starts at the beginning of the list, and goes through to the end, just in case there are forecasts with names which don't fit the pattern (i.e. if you saved them under a different name). After that, it will have found all forecasts in the time bracket you defined, and now begins going through the forecasts and listing them, including the net errors for each min and max temperature forecast. This may well take several minutes. In fact, if you may want to go find something else to do while it runs! If more than about 500 forecasts are listed, the text box will be almost full, and subsequent individual forecasts will not be displayed, though they are all written to the file details.txt. On some computers, this process may take up almost all the CPU, so that the fan may come on, and other programs will be slowed down a lot. Just be patient and let it do its thing!

Finally, the routine will finish, and you will be presented with about a page of analysis (though scrolling up can show up to 500 individual forecast results). Note: the entire analysis is saved in a file called details.txt, in your WXSIM folder. The example below is for a site in Finland. A wealth of information is shown, and you can also scroll up to see all the forecasts and results that went into the analysis. You may see some of the listed forecasts flagged (on the left) with the symbols M, I, G, >, or C. "M" means one or more data files were missing. "I" is similar, but means that – while the file exists – it was incomplete (for example, whole days missing). "G" indicates that, while most of the data were there, small gaps (i.e. an hour or two) have invalidated the data for inclusion. ">" means that, for at least one of the six forecast periods (min and max temperatures out to +3 days), the error exceed the user-defined threshold. Generally, one should not just "throw out" bad forecasts, but the idea here is that you set a really large value (i.e. 10 C = 18 F or more) and just about the only way WXSIM would ever miss that badly is if a major error occurred, such as ingestion of bad data. This is rare, but the option is available if needed. "C" flags and removes from the analysis forecasts with "bad" bias correction factors. This refers specifically to a bug that existed in wret.exe for a while, which allowed out-of-range slope factors to be used.



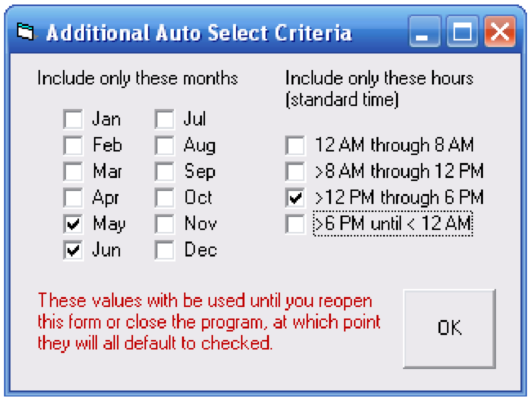
```
                    7.9  9.1 -1.2  21.4 21.7 -0.3   8.0  9.1 -1.1  21.5 21.6 -0.1   8.5  9.3 -0.7  22.3 21.8  0.5
Note: All forecasts are weighted equally.
Forecasts for Pertteli. Using Celsius.
Time to complete run = 10 min, 16 sec
Number of forecasts used = 96 (100% of total checked)
Day                    2             3             4                averages
Temperature         min    max    min    max    min    max       min    max    avg
Mean Net Errors:   -1.17  -0.31  -1.09  -0.09  -0.75   0.53     -1.00   0.04  -0.48
Mean Abs Errors:    1.83   1.67   1.92   1.64   1.58   2.23      1.78   1.85   1.81
RMS Errors     :    2.27   2.13   2.39   2.19   2.04   3.02      2.23   2.44   2.34
StDev of Errors:    1.94   2.11   2.12   2.18   1.90   2.97      1.99   2.42   2.20

Dew Point           mean   mean   mean
Mean Net Errors:   -1.22  -0.74  -0.25

Mean Temperature Error = -0.48     Mean Range Error =  1.05    Mean Dew Point Error = -0.74
Pred range = 13.61   Act range = 12.56  Prelim range mult = 0.923
Mean Temperature Error as Function of Departure from Normal: Intercept = -0.33   Slope = -0.12047

Temperature data, equally weighted, for complete days only.  Number of forecasts used = 96
              Min      Max      AM*      PM*    (Min+Max)/2 (AM+PM)/2  All Hrs   Max-Min   PM-AM
Pred:         8.14    21.74     9.44    20.29     14.94      14.86     15.36     13.61    10.85
Act:          9.14    21.70    10.40    19.92     15.42      15.16     15.29     12.56     9.53
Pred-Act:    -1.00     0.04    -0.96     0.36     -0.48      -0.30      0.07
Pred/Act:                                                                        1.084    1.139
Excess variability in min =  0.05; in max = -0.32
*AM is mean from 4:00 to 7:00 AM and PM is mean from 1:30 to 4:30 PM.

Previous Values of Correction Parameters:
 Temperature Offset (Intercept) = 0.27724 (raised temperature about 2.52 C)
 Diurnal Range Multiplier = 1.18575
 Temperature Correction (Slope) = 0.00212 (9.53% more pull towards climatology)
 Dew Point Offset = 0.08852 (raised dew point about 1.79 C)
Proposed New Values of Correction Parameters:
 Temperature Offset (Intercept) = 0.305 (would raise temperature about 2.78 C)
 Diurnal Range Multiplier = 1.09431
 Temperature Correction (Slope) = -0.00056 (2.52% less pull towards climatology)
 Dew Point Offset = 0.12506 (would raise dew point about 2.53 C)
Current MAE of 1.81 can be corrected to about 1.72, and with no previous corrections would have been about 3.05.
```

Maximum error (in max or min)
for inclusion (through day 4)
10
deg C
More Criteria
Show Graphs
Accept proposed values?
(see correc.txt for history)
YES
NO

Notice that the days are called 2, 3, and 4, instead of 1, 2, and 3. This is because this routine does not look at same day accuracy (except that you can view it to the extent the data is there, on the comparison graphs). This is because we may be mixing forecasts from 6 AM (which forecast the day's high) with ones from 10 PM, when the high is long past. So, day 2 is the first day shown and, on average, its minimum temperatures will have been forecast about 18-19 hours earlier, while the first maximum looked at is, on average, about 27 hours after forecast. In a way, that's too bad, because I think WXSIM has relative strength in short-term (less than 24 hours) forecasting, but can't document itself here with this routine. This should also be considered when comparing WXSIM's error (as shown here, not necessarily as shown in third-party scripts) to those of other sources; lead-time matters.

Something else that calls for some explanation: the "AM" and "PM" averages, and the "Excess variability" figures. WXSIM knows that temperatures fluctuate on a time scale (minutes or less) shorter than it (or any model) can forecast, and that extremes (min and max) for the day will lie somewhat off the relatively smooth forecast curve (or curve of half-hourly averages, used in wret.exe). Some sites have more natural variability than others. I included these data in order to study this variability, and have found that I indeed calibrated it pretty well, but some sites will show discrepancies that might need attention. Right now, though, the correction factors are based on forecast and actual extremes, not on the smoother diurnal curves.

Note also the 'More Criteria' button. Clicking this brings up a form that allows you to limit the analysis to selected months or time periods. You have to actually do an analysis first to access the button and the form, so that if you want to make special settings like this, you will actually need to complete another run. The changes stay in effect until you either close wret.exe or access the form (they are not saved to the program initialization file). These are intended to prevent accidental or over-use of a rather specialized option. The form looks like this:


Additional Auto Select Criteria

Include only these months

Jan [ ] Jul [ ]
Feb [ ] Aug [ ]
Mar [ ] Sep [ ]
Apr [ ] Oct [ ]
May [✓] Nov [ ]
Jun [✓] Dec [ ]

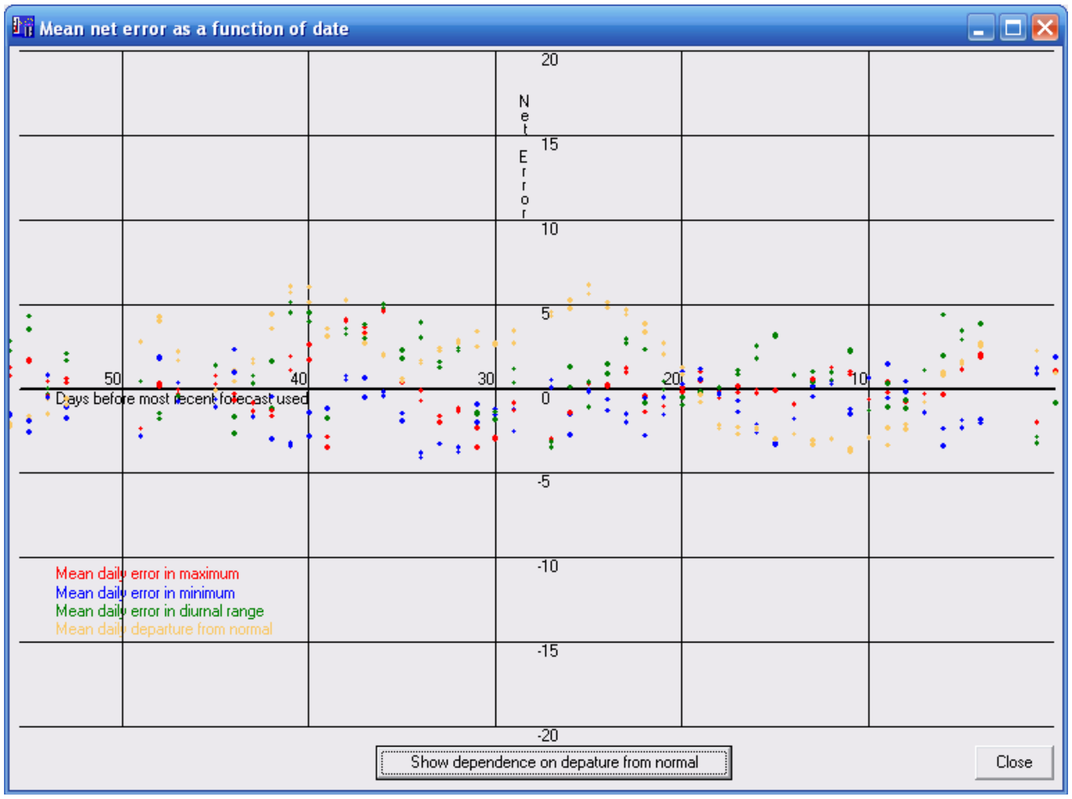
Include only these hours (standard time)

12 AM through 8 AM [ ]
>8 AM through 12 PM [ ]
>12 PM through 6 PM [✓]
>6 PM until < 12 AM [ ]

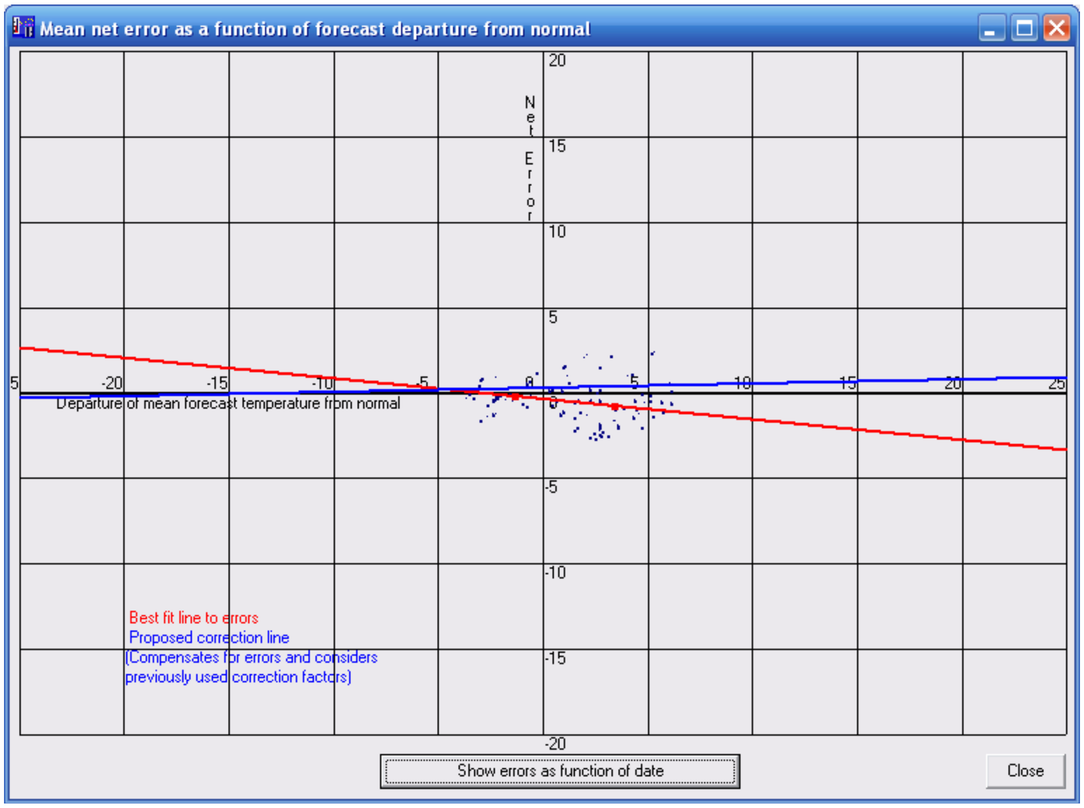
These values with be used until you reopen this form or close the program, at which point they will all default to checked.

OK

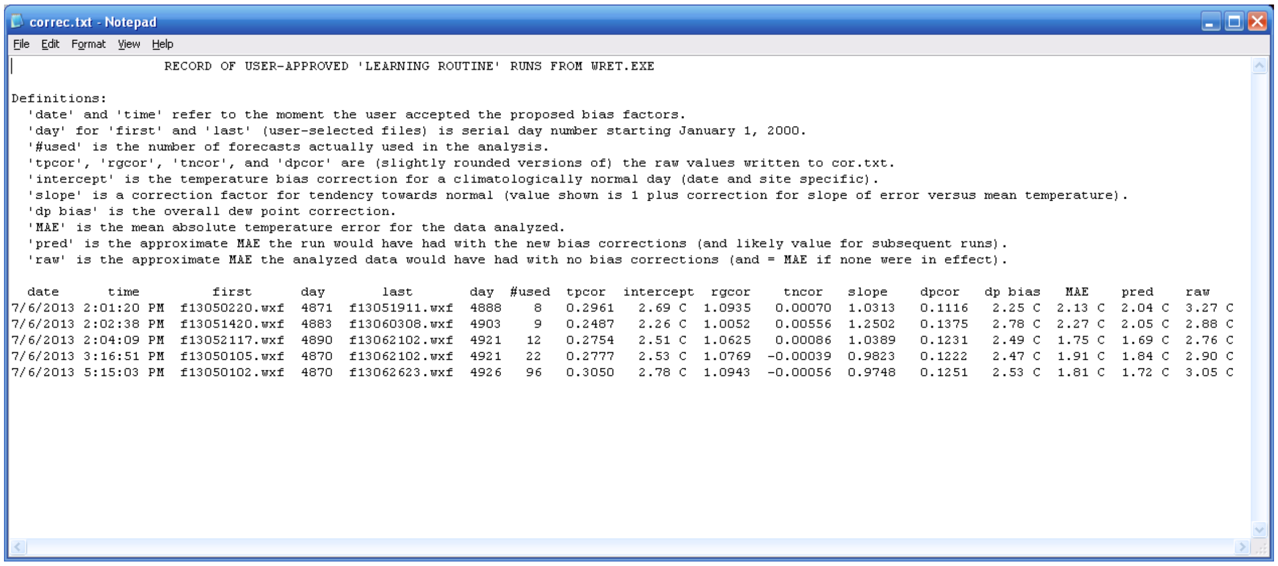
The 'Show Graphs' button enables two different types of display, between which you can toggle back and forth. The first is 'Mean net error as a function of date'. This shows mean net error for daily maximum, minimum, and average ((max+min)/2) , and also the departure of this average temperature from climatological seasonal normal for the site, all as a function of the number of days before the last forecast analyzed. It may be useful to see what types of weather patterns resulted in certain kinds of biases. There are periods, such as 8-9 days, and 17-20 days before the last forecast when WXSIM was very accurate and others, such as between 30 and 40 days before the last forecast, when it was having some trouble. (Scroll down to see image).



The other graph is 'Mean net error as a function of departure from normal'. This illustrates net error in slope-intercept form. The vertical axis is the net error, and the horizontal one is departure from normal (for the *forecast*, in this case, not the actual result). The red line shows a slight tendency for the cooler days were forecast too warm, and the warmer days to be forecast too cool. The blue line shows the proposed correction to this, as well as a very slight warming relative to the averages of the correction factors used in the analyzed forecasts.



A very recent addition (with a beta posted July 6, 2013) is a file archiving all the learning routine runs you've approved. The image below contains much of the explanation of the file, called correc.txt, and saved in the same folder with WXSIM. I think a very interesting thing will be to follow 'MAE', 'pred', and 'raw'. I think 'raw' will always be higher than the others, indicating that the learning routine is helping. I also think 'pred' will always be at least as good as 'MAE', indicating that the routine is indeed suggesting something which should be an improvement. However, improvements are likely to level off after a few months of use, and then will probably fluctuate. The next run's MAE may in fact not be as good as it was hoped ('pred'), but then on other occasions, the forecasts may turn out better than ever. I'm interested in feedback on the behavior of these figures over time, and this new file documents it nicely.



```
                   RECORD OF USER-APPROVED 'LEARNING ROUTINE' RUNS FROM WRET.EXE

Definitions:
  'date' and 'time' refer to the moment the user accepted the proposed bias factors.
  'day' for 'first' and 'last' (user-selected files) is serial day number starting January 1, 2000.
  '#used' is the number of forecasts actually used in the analysis.
  'tpcor', 'rgcor', 'tncor', and 'dpcor' are (slightly rounded versions of) the raw values written to cor.txt.
  'intercept' is the temperature bias correction for a climatologically normal day (date and site specific).
  'slope' is a correction factor for tendency towards normal (value shown is 1 plus correction for slope of error versus mean temperature).
  'dp bias' is the overall dew point correction.
  'MAE' is the mean absolute temperature error for the data analyzed.
  'pred' is the approximate MAE the run would have had with the new bias corrections (and likely value for subsequent runs).
  'raw' is the approximate MAE the analyzed data would have had with no bias corrections (and = MAE if none were in effect).

  date      time          first       day       last        day  #used  tpcor  intercept  rgcor    tncor    slope    dpcor   dp bias   MAE     pred    raw
7/6/2013 2:01:20 PM  f13050220.wxf  4871  f13051911.wxf  4888    8   0.2961    2.69 C  1.0935   0.00070  1.0313   0.1116   2.25 C  2.13 C  2.04 C  3.27 C
7/6/2013 2:02:38 PM  f13051420.wxf  4883  f13060308.wxf  4903    9   0.2487    2.26 C  1.0052   0.00556  1.2502   0.1375   2.78 C  2.27 C  2.05 C  2.88 C
7/6/2013 2:04:09 PM  f13052117.wxf  4890  f13062102.wxf  4921   12   0.2754    2.51 C  1.0625   0.00086  1.0389   0.1231   2.49 C  1.75 C  1.69 C  2.76 C
7/6/2013 3:16:51 PM  f13050105.wxf  4870  f13062102.wxf  4921   22   0.2777    2.53 C  1.0769  -0.00039  0.9823   0.1222   2.47 C  1.91 C  1.84 C  2.90 C
7/6/2013 5:15:03 PM  f13050102.wxf  4870  f13062623.wxf  4926   96   0.3050    2.78 C  1.0943  -0.00056  0.9748   0.1251   2.53 C  1.81 C  1.72 C  3.05 C
```

How to Easily Make a Combined Plot

Before leaving the subject of the learning routine, I'll explain here a much easier way to do the nice combined comparison graphs – without having to click dozens or hundreds of times!

First, use Comparison to Actuals to make a simple, one-forecast plot (this should be the first in the series of forecasts you want included). Then check 'Combine data' in the lower-right part of the form. Close that form (using as, always, the Close button, and NEVER the little red Windows "X" in the upper right).

Next, act like you are going to do an Auto-Select routine, by clicking on the next forecast (in the list, right below the one you just did) and then clicking Set as Earliest Date. You might get a message saying that, if you want to do a learning routine run, you need to choose 4 days; you aren't restricted to this for graphing purposes, though. Then, scroll down to the *second-to last* forecast you want included, and click Auto Select. You will be warned that data are being combined, in a way that makes this inappropriate for getting correction factors (and it won't give you any anyway in this case). That's OK, because that's not what this is about; we just want an easy way to make a combined data graph. So, click OK.

The program will go through steps as if it is trying to get correction factors, but if you look closely at the forecast lines as they scroll by, you'll see that the errors seems to stabilize, and the temperature start looking all about the same. This is because the data are being cumulatively averaged. It may actually be interesting to watch – if you have nothing else to do!

When this is done, click 'No' (the only choice you have) to leave the form, and then look back at the list of forecasts. You will probably find that the one below the last one you clicked is highlighted (if not, you can click on it now). Simply click Comparison to Actuals one more time, and you will have your combined graph! You can't directly save the graphics, but you can use <Alt><PrtScrn> to grab the front form as a screen shot and then save it into Word or Paint, for example.

You might try experimenting with the ‘More Criteria’ button, to make the plot specific to certain months or times of day. You will generally find that forecasts made at later times of day make the curves disappear up to that point – appropriately, as the earlier-in-the-day forecasts can’t be averaged with empty data. Also, if you pick one that’s too recent, so that not all the days forecast have occurred yet, this will truncate the right side of the plotted data (again, as it should be). Occasionally, strange spikes may appear, indicated gaps in data, which aren’t as well screen out here as in the learning routine itself.

Finally, watch out for these two issues: (1) the Daylight Saving Time setting in wret.exe must correspond to that used by your weather station software at the time of the forecast(s), to make things line up right, and (2) if you are using Weather Display, and have it set to output kilometers per hour, you are actually storing the data in KNOTS in the log files, and should select that as the units for display in wret.exe, temporarily! You are still likely to find your measured wind speeds below those forecast, because it’s very difficult to mount home anemometers to standard (10 meters above ground, and no obstacles for something like 200 meters all around!).

One more technical note: wind direction in wret.exe (and in WXSIM) is averaged in a rather sophisticated, and appropriate way. All the wind speeds and directions are considered as vectors, and vector addition is performed, followed by dividing by the number of observations, to get a “resultant” wind direction. This is much superior to simply averaging the wind directions (which then tend to be pretty much south!) and also to getting a “dominant” direction by some algorithm to count how many times it was from a particular direction. In WXSIM itself, a resultant speed is also shown, but in the present case in wret.exe, the average speed is just the average of the speeds themselves.

OK, the correction factors need some explaining here. Afterwards, I will say more about how the ‘pred’ and ‘raw’ figures are generated.

The Correction Factors and What They Do

It’s worth noting that these correction factors are of a different nature from the changes I make to cty.fdt files to effect desired changes. I tested their effects carefully, but it’s been difficult to establish with confidence things like how much data is enough but not too much, whether to weight recent and older data equally, and how much to constrain the possible ranges of these factors. In most cases, the routine seems to be helping improve accuracy, especially for sites where I didn’t get the original customization quite right (despite sincere efforts!). In some other cases, I suspect that undetected bad data and/or blown cloud cover or wind forecasts may have thrown the rouinte out of kilter and actually degraded forecast accuracy. This is an issue I am working on, and the performance should be improving over time, and helping improve forecasts more often than not.

It is important to understand the variables which are affected by the factors generated by this routine. There are four: (1) the overall temperature bias, (2) the diurnal range factor, (3) the tendency towards normal factor, and (4) the dew point bias.

The first is fairly simple. It is aimed at correcting overall temperature bias – day, night, summer, winter, cold spells, hot spells being affected more or less equally. It basically adds or subtracts something from all the temperatures. However, it should be noted that this is buried in the model itself, and is not a straightforward post-processing issue. For example, I wouldn’t want to have the forecast saying rain, and have an above-freezing temperature post-processed to sub-freezing! This factor is allowed to have values from -0.305 to +0.305, corresponding roughly to lowering or raising (respectively) overall temperatures by about 5 degrees Fahrenheit (2.78 Celsius degrees).

The second is fairly straightforward, but a bit more complex. Diurnal range is the difference between daily low and high temperatures. To increase this implies lowering the lows and raising the highs, by equal amounts (as (1) took care of overall bias). However, these amounts are in terms of multiplicative, not additive factors. For example, this would not lower all lows by a degree and raise all highs by a degree. Instead, it stretches or shrinks the whole curve, perhaps making the range 1.2 times what it otherwise would have been. A day with a low of 20 and a high of 40 would become 19 and 41, but one with a low of 25 and a high of 35 would become 24.5 and 35.5. Once again, this is NOT post-processed, but built into the model from the start. This factor is allowed to range from 0.80 to 1.25, meaning the diurnal range will be from about 80% to about 125% of the value it would have otherwise.

The third one is the hardest to describe. I have found that a type of error that can occur is a tendency to be either too "careful" or too "bold" regarding excursions far from normal temperature. For example, too much "careful" might mean that a day which was forecast to have a high of 10 might really end up as 7, while a day forecast to be 40 ends up as 43. Too "bold" would result in unrealistically extreme temperatures being forecast, in either direction. This correction factor, if positive, makes cold (for the season) temperatures warmer and warm temperatures cooler, and of course the opposite if negative. While this is an important correction to make, some types of climate or weather patterns make it very hard to accurately establish. In particular, rather maritime climates, or times of year with monotonous temperature, give it little to go on. A cloudy day in summer might be cool, but is not really the kind of thing this routine is looking for. Because of the volatility of the resulting factors, users often see message boxes that this factor (or its initial calculation, anyway) is "out of range and will be limited". The best results for this factor come from including a substantial period of data (a few months) including a changeable time of year, with a hint that the result may be good given by the factor staying well in range, with no need to be limited. This factor may range from -0.00556, which makes temperatures be about 25% further from normal than they would be otherwise, to +0.00556, which makes temperatures about 25% closer to normal than they would be otherwise.

The fourth factor is the only one not involving temperature directly. This one affects overall dew point bias. Positive values raise the dew point and negative ones lower it. As with all the other factors, this is NOT post-processing. This factor is allowed to have values from -0.1375 to +0.1375, corresponding roughly to lowering or raising (respectively) overall dew points by about 5 degrees Fahrenheit (2.78 Celsius degrees) – except, of course, that the dew point is not allowed to exceed the temperature in WXSIM (or reality, for the most part).

A considerable coding difficulty was the fact that these factors are not independent in WXSIM! A fairly easy-to-see example is the relation between diurnal range and tendency to normal. A day with a really big diurnal range could of course have an above-normal high and a below-normal low. This could also be interpreted as a freedom to stray far from normal! Clearly, these two factors "bleed over" into each other, and I've made careful adjustments to have them act independently as far as end results go. For example, if the tendency of normal is made low (negative), the diurnal range is reduced to isolate that effect. Likewise, higher dew points make lower diurnal temperature ranges. Less obviously, lower overall temperatures make slightly lower diurnal ranges. I've tried from the outset to keep this all straight, and I pretty much have, but it warrants periodic review and possible tweaking.

A possible advantage of this "learning routine" over a one-time re-customization could be that it could take into account changing conditions over time. One example might be changes in land-use in the area of the station. More likely examples are simply things like changing seasons and stubborn weather patterns that are out of the ordinary. There is a difficult balance to strike, however. On one hand, we want as much data as possible to have statistical validity for our corrections. In my understanding, the errors in these kinds of things scale as the inverse square root of the number of data points (forecasts); errors due to insufficient data using 50 forecasts could be cut in half by using 200 (four times as many). On the other hand, 200 forecasts might span more than a season, and would certainly be too long to catch situations like a blocking anticyclone causing unusual weather for two weeks or so.

I have not managed to come up with a one-size-fits-all solution to this problem of determining how many forecasts (or days) should be used in the analysis. I do have some impressions, though. If WXSIM is doing a good job with the cloud and wind parts of the forecasts, so that most temperature errors are generally NOT due to "blown" cloud or wind forecasts (such as we saw in the first screen shot, above), then a shorter time period can be used. For example, in a desert climate, where the cloud forecast is almost always right, biases might be dependably corrected with as little as two weeks of forecast-actual comparisons. However, where and when weather is very changeable and difficult to forecast, a lot of data is needed to "see through" the noise (OK, I just mixed senses metaphorically!). In the example above, a month may have been enough. However, the errors are already small, and could be an "artifact" of some odd, random stuff; so, even in that case, I think 2 or 3 months might be called for. More than that, though, would blur together seasonal differences would like to catch. The best case for using a short period of data is if large systematic errors are immediately seen, in spite of accurate cloud and wind forecasts.

A practice which should probably be avoided is a “reactive” one, in which the user decides to run wret.exe to get correction factors because of a particularly bad spell of forecasts, and then includes only those forecasts in the analysis. First, while this allows WXSIM to learn from its mistakes, it denies it the chance to learn from its successes as well. The result is likely to be a temporary improvement in the forecasts, followed by a sudden change to worse-than-ever forecasts once a change in weather patterns occurs. An indication that this is happening is if the correction factors jump around from one end of their allowed range to the other. The allowed ranges provided are already on the upper end of what I think should ever be needed, assuming the customization was reasonably good.

I have some concern that using the learning routine on a relatively poor initial customization may cause instabilities or unexpected problems. It’s better if the biases are already fairly small, so that the correction factors are well-behaved and predictable in their results. For this reason, I recommend users with relatively large forecast errors contact me for other solutions first, because trying to have the learning routine handle it all. Once the customization is reasonably good, the learning routine can take it from there, from then on.

How to Know if the Learning Routine is Helping, or How Much it Could Help if You Were Using It

Forecasts will of course never be perfect. Furthermore, there is often a problem of diminishing returns. It seems the first half of your effort brings 90% of the results, and the rest you really have to fight for! In WXSIM’s case, there are inherent limitations in the internal model itself, but also (as discussed at the beginning of this article) the fact that it must rely (in auto mode, at least) for many non-temperature things on external model data, from GFS or NAM, which – while quite good and generally improving year-to-year – are not perfect, and are not even the best out there anyway. The European model, ECMWF, tends to beat the GFS (probably more significantly in the medium-long range) and mesoscale numerical models like the WRF often catch details blurred out by the GFS’s 0.5 by 0.5 degree data grid (and internal resolution). Right now, the only way to get these models’ output into WXSIM is to purchase the expensive ECMWF data (or run the WRF on a powerful PC, for hours) and then manually click the data into WXSIM using the Interrupt Planner. This is not practical for most users.

As we have seen, error tends to be separable into two parts: “random” (mostly meaning due to information unavailable to us) and “systematic”. The former is indicated in wret.exe’s analysis largely as standard deviation, while the latter is in the form of the bias correction factors discussed above. Fortunately, with the four different types of bias (three for temperature and one for dew point) treated here, we can go a long way towards eliminating systematic error in WXSIM. The question now becomes, “if this is successful, how much does it really help”?

This is not a simple (or certain) thing to figure out, but I recently made a major effort to do so, and learned a lot in the process. Part of what I did was to create an Excel spreadsheet with a Monte-Carlo simulation of 1000 forecasts, with a close approximation to a Gaussian (normal) distribution of errors. I also did about 20 instances of it, simulating 20,000 forecasts, with errors. I also used smaller sets (60-120 forecasts) which I "randomly generated" myself, AND tested everything on data sets from a few different users.

I developed empirical equations for the three separate (somewhat, at least) temperature factors: the overall temperature bias ("intercept", the tendency towards normal ("slope") and the diurnal range factor correction. I am assuming (probably not quite correctly, but I think it must be close) that these operate independently. The three types of data I used (forced Gaussian, generated by me, and actual users' data) each imply slightly different parameters, but they're similar, I think, and the mix of sources probably makes this pretty robust.

Here's an example of what I found: The amount of improvement in mean absolute error to be expected from correcting a simple "DC offset" (sort of EE terminology there) temperature bias is a factor of about 1.43 (and I wonder if maybe it's really the square root of two???) if the bias is equal in magnitude to the standard deviation ("population" version used here, but the numbers are large anyway), and is proportional to about the square (I actually used the 1.83 power, as determined empirically) of the bias as a fraction of standard deviation.

Also I found that the effects of the diurnal range correction grow as at least the square (I used 2.3 power) of the difference from 1, and also as at least the square (2.25 power here) of the actual average range itself. Users with dry climates and big ranges get a pretty good boost in accuracy by correcting this.

I even managed to investigate the effects of the "slope" factor. These appear to grow more slowly with the size of the factor (1.25 power), and probably also with the season, but I was "up to my ears" (or "over my head") at that point, and dealt only with a "season" in which standard deviation of daily temperature is a bit under 6 F degrees (somewhat over 3 C degrees). I think it should matter more in more extreme seasons, so the benefits of getting this right will likely be bigger than I indicate in this new output (including the 'pred' and 'raw' figures in the new correc.txt file).

Concluding Remarks

WXSIM has been a major, ongoing project of mine for over 30 years as of this writing. What began as a playful physics-based modeling experiment has become, an remained a serious forecasting tool. Efforts to improve it necessarily entail honest and accurate evaluation of its strengths and weaknesses, and the tools described above provide a way to both learn what these are and act on them to improve forecasts. The better that users understand them, the more useful they are likely to be.

A nearly-forgotten aspect of WXSIM, though, is that it's supposed to be a tool to *help* the user produce the best forecast possible. This does not necessarily mean that WXSIM, on its own, will beat all other forecasts. It *does* imply the hope that WXSIM will bring something new and independent "to the table". I always felt this to be the case, but recently did a very specific study, with interesting results.

I used some comparison data from a long-time customer's (Oebel Reitsma) web page, using Jim McMurry's script and pasted it into a spreadsheet. I used 233 forecasts (going back further seemed to include some problem that made WU's standard deviations too big), spanning last October until now. I found that Weather Underground (see more on them below) had a 4-day average MAE of only 1.667 degrees C, while WXSIM's was 1.856. That's the "bad" news (actually those are both very good numbers!). The good news is that, when I did a weighted average of the two (counting WU 62% and WXSIM 38%), the error dropped to 1.558! This is including the decimal places, but even if rounded off to a whole number, the error is still just 1.573.

(By the way, in WXSIM's favor in Oebel's case, he does use the learning routine - on the other hand, his location is topographically simple, so WXSIM customization doesn't give quite the advantages it might in a more complex location. I suspect his situation is typical of many WXSIM users. Also, he has since told me that he was using the learning routine in a "reactionary" way, which I do not think produces the best results).

WU advertises that their "Best Forecast" system does a custom blend of various model data (definitely including ECMWF), and data from their network of about 25,000 reporting stations. They have optimized about as well as they can, and the result is quite good. On the "two heads are better than one" assumption, one would generally think that another "opinion" might help, but of course if it's a bad opinion, or even if its a good one that's already in the optimized mix, further averaging will in fact make the result worse. However, if it's a new and different opinion, that sometimes really knows "what it's talking about", taking it into consideration will indeed help.

This is what seems to be happening with WXSIM. I have sometimes been concerned that it takes GFS data into account so much that it becomes largely a "clone" of it. However, if that were really the case, mixing GFS into an already optimized mix with GFS wouldn't help. In fact, it appears something about WXSIM's native, local modeling abilities is "shining through" and contributing to an improved forecast. Of course, in some (many, I hope) cases, WXSIM is the better forecast anyway. Even then, though, mixing in WU (or some other really good forecast) should help, as well. If, on the other hand, one of the sources is really bad, then mixing the worse one in will of course hurt. The good thing is that, even when WXSIM is not quite as good on its own, "listening" to it will improve things.

Now what I'm thinking is that script developers might consider some method by which users could create a better-than-ever consensus type forecast. It could be pretty sophisticated, as the comparison script might include another column, with a weighted average, and there could be tools to control the weighting. One person might be getting relatively disappointing results with WXSIM, maybe with errors of 2.5 degrees while WU is only 2.0, say, but combining them maybe 80-20 WU-WXSIM might improve things to 1.95. The reverse might also be the case for some. Then there will be many cases where the two are quite close. Perhaps if they are tied at 2.0 each, combining them 50-50 might result in 1.85 or better.

Also, sometimes this strength depends on the time frame and whether it's a min or a max temperature forecast. The script might include ways to evaluate and assign these separately. In Oebel's case, WXSIM was winning at short-range max temperature forecasts, in which case maybe the weighting should be 60-40 WXSIM-WU for those situations, but something else for others.

I hope this information helps you enjoy and benefit from WXSIM more than ever before. I'm very interested in feedback.

Happy forecasting! ☺

Tom Ehrensperger

July 6, 2013