

FVS Made Easy

Perry C. Shafran

Updated 20 March 2007

The NCEP/HPC-EMC Forecast Verification System (FVS), a graphic verification tool developed and maintained by Keith Brill (HPC), can be used to plot verification of many different items. In the NCEP Verification System Documentation paper, the system used to create the “partial sum” files, or the “VSDB” files, was described in detail. The FVS system searches VSDB files, computes verification performance measures, and displays results. This paper describes many ways that the user can take advantage of FVS's options to quickly plot nice-looking graphics and create a wide array of verification products. Also, one can get the printout of the values that go into the graph. Skip to Page 31 for details on how to do this.

This particular document is a living document. That means that when new capabilities are added to FVS they will be added here.

FVS Setup

FVS is installed on the workstation system, mist, dew, and haze. In order to set up your working environment to be able to use FVS, these lines need to be added to the .cshrc file or the .profile file.

On the workstation system, add the following lines to your .cshrc.

```
setenv VSDB ~wd22kb/vsdb
setenv VSDB_TBL $VSDB/tables
setenv VSDB_HLP $VSDB/help
setenv VSDB_SCRIPTS $VSDB/scripts
setenv VSDB_DAY1 199001010000
setenv VSDB_DATA /mmb/wd20ps
source $VSDB/scripts/for_fvs
```

The VSDB_DAY1 is not necessary; it only indicates the first date of available VSDB files. The VSDB_DATA variable denotes where your VSDB directories are. Please consult the NCEP Verification System Documentation paper for details on requirements of VSDB directories.

On one of the IBM supercomputers (mist/dew) add the following to the .cshrc file:

```
setenv VSDB /u/wx32kb/vsdb
setenv VSDB_SCRIPTS /u/wx32kb/vsdb/scripts
source $VSDB/scripts/for_fvs
source /nwprod/gempak/.gempak.csh
```

Or to the .profile:

```
export VSDB=/u/wx32kb/vsdb
export VSDB_SCRIPTS=/u/wx32kb/vsdb/scripts
. /nwprod/gempak/.gempak
. /u/wx32kb/vsdb/scripts/for_fvs.sh
```

The above items will shift to a different directory in the future. On haze, the research computer, replace wx32kb in the above two with wx20ps.

With the above FVS is able to run on the appropriate system. Users should also set the environment to display to their desktop computer so they will be able to get the X-windows that FVS puts out. For any help on using FVS, the user can type “fvs h” to get information.

The trace.ctl file

Generally one types “fvs u” on the command line to set up the parameters needed to create various types of plots. The “fvs u” uses information in a set of “.” files in the VSDB directory at the level of the model directories. A script named \$VSDB_SCRIPTS/update.files is provided to update these “.” files. Those responsible for populating VSDB directories must run update.files periodically. This supports the use of “fvs u” to make it fairly easy to create trace.ctl files.

By doing so, a trace.ctl file is being created. It is highly recommended that the user create his various control files by using fvs u and following the menu-based instructions. Editing a trace.ctl file is tricky and generally should be limited to minor changes.

The “fvs u” option asks a series of questions regarding how the user wants the data plotted. All of the options discussed below can easily be set in the “fvs u” program. Once “fvs u” is run, the control file can be used and edited to plot the same statistics repeatedly for different data. One could even place the control file in a script and have some of the conditions created on the fly.

However, it is not necessary to run fvs u each time to create a plot. Many users would prefer not to run the job manually and prefer a method to be able to produce these plots in a scripted job. The trace.ctl file can be edited if, say, the items that need to be changed are a parameter, or a level, or a time period. Below, some examples of control files are presented to help users make it easy to make a plot without going through the time-consuming process of setting up the parameters in fvs u.

Time series

Here is an example of a control file one would use if creating a time series:

```

1 5 1 2 1 1 1 0/0/1/0
15 0 0 200611010000 200611100000
2 5 1 2 1 1 1 0/0/1/0
15 0 0 200611010000 200611100000
1 1 0 2 1 1 1 0/0/1/0
18 0 0 V01 0
2 1 0 2 1 1 1 0/0/1/0
18 0 0 V01 0
1 2 0 2 1 1 1 0/0/1/0
18 0 0 NAM/218 0
2 2 0 2 1 1 1 0/0/1/0
18 0 0 GFS/212 0
1 3 0 2 1 1 1 0/0/1/0
9 12 0 0 0
2 3 0 2 1 1 1 0/0/1/0
9 12 0 0 0
1 8 0 2 1 1 1 0/0/1/0
18 0 0 ONLYSF 0
2 8 0 2 1 1 1 0/0/1/0
18 0 0 ONLYSF 0
1 9 0 2 1 1 1 0/0/1/0
18 0 0 G236 0
2 9 0 2 1 1 1 0/0/1/0
18 0 0 G236 0
1 10 0 2 1 1 1 0/0/1/0
18 0 0 SL1L2 0
2 10 0 2 1 1 1 0/0/1/0
18 0 0 SL1L2 0
1 12 0 2 1 1 1 0/0/1/0
18 0 0 T 0
2 12 0 2 1 1 1 0/0/1/0
18 0 0 T 0
1 14 0 2 1 1 1 0/0/1/0
18 0 0 SFC 0
2 14 0 2 1 1 1 0/0/1/0
18 0 0 SFC 0

```

With the example control file above, one may use this file to plot a time series. The items above are, in order as they appear in the control file

- 1) Starting/ending date of the period to be verified
- 2) The version number: this is always V01
- 3) The models to be verified. In our example, the models are NAM/218 and GFS/212, meaning that this is how these items are written in their respective VSDB files. In this example the model datasets are the NAM grid 218 and the GFS grid 212. This is the only example, in this particular case, where the two examples do not match each other. That doesn't have to always be the case, as noted below.
- 4) The forecast hour. In this case the 12-hr NAM is being compared to the 12-hr GFS. However, these two numbers do not have to match. One could verify, for example, the 24-hr NAM compared to the 12-hr GFS, if so desired.
- 5) The verifying dataset used. In this case they are both ONLYSF, which represents surface data.

- 6) The verification area. In this case both are using G236, which is defined as the RUC output grid. This simply states that only observations contained within the G236 domain are part of the verification output.
- 7) The statistic type. In this case the type is SL1L2, which represents items like root mean square error, bias, correlations, etc. How to plot these in FVS will be described in further detail later. For wind, the data type would be VL1L2.
- 8) The variable. In this case T is being used, although it could easily be SLP, RH, or some other item.
- 9) The level. In this case it is SFC (which is 2-m data for T and RH, 10-m data for VWND).

Most of the numbers in the header line the user does not need to worry about. However, there are a few items in each line that might be of interest. Let's take an example:

```
1 5 1 2 1 1 1 0/0/1/0
```

Here is what each item in the header line represents:

- 1) Trace numbers. The max number of traces is 24.
- 2) Search parameter number
- 3) Variable type (1 for independent and 0 for dependent)
- 4) Data combination code. Must be 2.
- 5) Number of conditions following header. Max is 64 for conditionally determined independent variable entries, 32 for a dependent variable search list.
- 6) Plot number to which trace belongs
- 7) Plot type
- 8) Data consistency checking flags

The 2nd number in each row represents the number of the fvs u selection, if one were to use fvs u to create the trace file. Here is a list of what the 2nd number should represent:

- 1) Version number (always V01)
- 2) Model being verified
- 3) Forecast hour
- 4) Forecast time windows
- 5) Verification time (YYYYMMDDHHNN)
- 6) Verifying time windows
- 7) Time increment for data

- 8) Verifying analysis
- 9) Verifying region
- 10) Statistic type identifier
- 11) Threshold for data
- 12) Parameter name
- 13) Level identifier
- 14) Level value
- 15) Verifying HHNN screen (whether this is a 00Z or 12Z initial time)

The 3rd number in each row represents whether the variable in question is a dependent or independent variable. A “1” in that slot represents an independent variable, while an “0” represents a dependent variable. In the time series, the dates selected are the independent variable, the others are dependent variables. Independent variables determine “bins” along the abscissa of the expected plot.

The 5th number in the row represents the number of conditions following the header. In the example above, note that it is always 1, and only one item follows each header. However, say that a plot has traces containing the statistics of several models, and the average statistic of the several models are desired. Then a change is required:

```
1 2 0 2 1 1 1 0/0/1/0
18 0 0 NAM/218 0
```

could be changed to, for example:

```
1 2 0 2 3 1 1 0/0/1/0
18 0 0 NAM/218 0
18 0 0 GFS/212 0
18 0 0 RUC/252 0
```

By making this change fvs will then read all the partial sums for all three of these models and create the average statistic for all three models. This is different than having these in separate sections, which creates three traces:

```
1 2 0 2 1 1 1 0/0/1/0
18 0 0 NAM/218 0
2 2 0 2 1 1 1 0/0/1/0
18 0 0 GFS/212 0
3 2 0 2 1 1 1 0/0/1/0
18 0 0 RUC/252 0
```

Users should use extreme caution in editing a file this way. If users are not careful in their edits, such as the proper numbering of traces or plots, then there is potential to create a trace.dat file that is unplotable. For example, if the 5th element is not changed from a 3 to a 1 in the above example, the fvs plotter will fail.

This capability exists for all sections of the control file. One may do statistics over several areas. For example, to do the verification for only the West Coast of the Continental US:

```
1 9 0 2 2 1 1 0/0/1/0
18 0 0 G104/NWC 0
18 0 0 G104/SWC 0
```

This assumes, of course, that the VSDB files themselves contain data with these specific areas. If the VSDB file does not contain, for example, any data verified in the G104/NWC or G104/SWC domains, nothing will be plotted. Of course always be sure that the 5th element of the header row properly contains the number of conditions to follow the header.

The 6th item in each row represents the plot number to which a trace belongs. A trace.ctl file need not contain the information for only one plot. One may have information for many plots. This number would need to be incremented for each plot. Keep in mind that, though, that if the user desires to have more than one plot in a control file that information such as verification time, level name, etc, would have to be repeated. It cannot be assumed that the information for plot 1 will be used for plot 2.

The 7th number is the plot type. This number is 1 for a time series, 2 for a scatterplot, 3 for a skill score vs. thresholds plot, and 4 for a categorically binned data plot.

The final four numbers, separated by a slash (/), are consistency checking parameters. The first number represents whether consistency checking is used. If that number is 0, no checking is used. A 1 in this slot tells the searcher that inter and/or intra-trace counts are consistent. A 2 in this slots eliminates all points that are not found in all traces. The third number tells the searcher whether or not to use FHO values that contain all zeroes for information (see later in this document for information on plotting FHO statistics). The fourth number represents the percentage of data loss to be allowed from consistency checking before the checking routines are relaxed. This is only used if the 2nd number in this group is 0. Quite often, this group is not used, though it can be very important to have consistency checking for many types of statistics for fair comparisons.

Many of the items above are not in this example of a control file. Also, the order of items in the control file is not important. If one were interested in plotting verification for 3, 4, or even more models, more lines would need to be added.

In the directory that contains your control file, then type “fvs s” in the command line. Be sure that the \$VSDB_DATA environment variable points to the proper database. If it does not, the searcher will not work. The “fvs s” command does the searching of the verification database and creates a trace.dat file, which is used for the plotting. Then type “fvs p” to begin the plotting.

After “fvs p” is done, this is what is seen:

Here are the contents of the trace data file:

Number of plots available = 1

PLOT # 1 is a time series composed of 2 trace(s).

INDEPENDENT VARIABLE FOR THIS PLOT:

VYMDH

STAT = SL1L2

GENERAL LABELS

#	FLD NAM	LABEL
1	VRSN#	V01
2	FHOUR	12
3	V_ANL	ONLYSF
4	V_RGN	G236
5	STATS	SL1L2
6	PARAM	T
7	LEVEL	SFC

STAT = PARAM = T

TRACE # 1 LABELS

#	FLD NAM	LABEL
1	MODEL	NAM/218

STAT = PARAM = T

TRACE # 2 LABELS

#	FLD NAM	LABEL
1	MODEL	NAMY/218

Select number of the plot to display (0=exit):

This is what is will be referred to as “Menu 1”. For the rest of this paper, the reader should return here for references to the various menus so they are not repeated. Hit “1” and then this is what appears:

Select plot display type by number:

- 0 - Print statistics for traces in plot defined above.
- 1 - Graphical display of plot as defined above.
- 2 - Scatter plot (Requires 2 series of computational results, either from the same trace or two different traces.)
- 3 - Old-style histogram (requires exactly one trace)
- 4 - Display scatter plot data as line plots
- 5 - Display ROC curves (select comp code 1032 below)

Enter choice:

This will be referred to as “Menu 2”. To do the time series, hit 1 and return, and Menu 3


```

*      1102          Compute RMS error (S[S]AL1L2, SL1L2)          *
*      1103          Compute Correlation (S[S]AL1L2, SL1L2)       *
*      1104          Compute Covariance (S[S]AL1L2,SL1L2)        *
*      1105          Compute Forecast Variance (S[S]AL1L2, SL1L2) *
*      1106          Compute Observation Variance (S[S]AL1L2, SL1L2) *
*      1107          Compute Forecast Std Dev (S[S]AL1L2, SL1L2)  *
*      1108          Compute Obs Std Dev (S[S]AL1L2, SL1L2)      *
*      1109          Compute Error Std Dev (SAL1L2, SL1L2)       *
*      1110          Compute BIAS + Error Std Dev (SAL1L2, SL1L2) *
*      1111          Compute BIAS - Error Std Dev (SAL1L2, SL1L2) *
*      1112          Compute BIAS**2                               *
*      1113          Compute -2*Covariance                        *
*      1114          Compute Error Variance                       *
*      1115          Compute Mean Squared error                  *
*****

```

Select computational code from table above. Please enter 3 (one for each trace) computational code(s).

OR

Select computational code from table above. Please enter one computational code(s).

Here we have a list of what options to plot. Note that this last line is different depending on rather you chose 1 or 2 in Menu 3. If you choose “1” in Menu 3, then you only type one of the above choices, which will be applied to all your plots. Typically, one would choose from the 1100 group above. For example, typing “1102” here would plot a line series of the 12-hr surface temperature RMS error.

If the user chose “2” in Menu 3, then this would produce a plot for each trace. For example, if obs vs. model is desired, using the example above where the user typed “1 1 2”, then here the user would type “3 2 2”. This means that choice “3”, for a plot of the observations, will be applied to trace 1. Then each “2” would produce a plot of the model for trace 1 and trace 2.

After this option is selected, two other options are seen. Generally they are not important except for advanced users, and one should simply press return after each choice.

```

Enter the intra-trace data combination lag interval (the
statistic value or score at a point is computed after
combining data from the preceding (LAG - 1) points with
data at the point)

```

```

Enter LAG (RETURN or 0 = no lag interval):

```

```

Enter # of trace points for running average smoother
window (RETURN or 0 = no smoothing).

```

Finally we get to Menu 5, the final menu:

```

*****
* The information in the control file is key word tagged. An equal *
* sign (=) separates the keyword from control information following *
* it. Here are the keywords and their meanings: *

```

```

*
* BORDER = Background lines color/dash/width
* CLEAR = Clear screen flag (yes or no)
* COLORS = Color number assignments
* DEVICE = Device|name|x size;y size|color type
* FACTOR = Scaling factor(s) to apply to data (lft_sf/rht_sf)
* HSTCLR = Histogram bar colors for each trace separated by ;
* L_TEXT = Text size/font/width/hw flag for axis tic-mark labels
* LABLEV = label level
*         0 = No labels
*         1 = Mix of user-defined and default labels
*         2 = ONLY user defined title and trace key labels
* LINE = Line colors/types/widths/labels
* MARKER = Marker colors/types/sizes/widths
* PANEL = Panel location (0=full screen)
* PTYPE = l.s. ; r.s. Y axis plot types/h:w ratio/lm;bm;rm;tm
* REFLIN = Reference line: x1,y1;x2,y2;x3,y3/color/type/width
* REVERS = Yes/no flag to switch X and Y axes
* RSTRCS = Right-side Y axis display trace #s separated by /
* RWTTLN = r.s. Line colors/types/widths/values for witness lines
* RYAXIS = Right side Ystrt/ystop/yinc/lbl;gln;tck
* T_TEXT = Text size/font/width/hw flag for title|trace|axis text
* TITLE = Title string
* TRACEn = User provided text label key for trace number n
* WITLIN = l.s. Line colors/types/widths/values for witness lines
*         If VALUES are missing, 0 is assumed.
* XAXIS = Xstrt/xstop/xinc/lbl;gln;tck
*         t_start:t_stop:increment/lbfq;gdfq;tkfq
*         For a time axis X is in days, starting at 0.
* XLABEL = Label text for X axis/# of char size units from margin
* XUDLBL = list of labels for X axis tic marks separated by /
* YAXIS = Ystrt/ystop/yinc/lbl;gln;tck
* YLABEL = Label text for Y axis/# of char size units from margin
* YRLBEL = Label text for r.s. Y axis/# of char units from margin
*****

```

The table above describes the graphics control file.

Below are the default entries in the graphics control file:

```

DEVICE = xw|VSDB
CLEAR = yes

```

Set a graphics control file entry:

Keywords may be abbreviated to first 3 letters.
Last entry overrides previous ones for a keyword.
Reasonable defaults are provided for unset keywords.

OR enter one of the following commands

```

r = run plot program with current keyword settings
h = display keyword help list
l = list current settings
e = exit this program

```

a followed by the name of a file containing keyword settings to append to the current file
f followed by the name of a file containing all keyword settings
h followed by a keyword gives help for that keyword
h colors gives help information on colors
s followed by a file name in to which to save the current keyword settings

The top part lists many of the graphics control options that are available to change the appearance of the plot. For help on any one, you may press h and that option (for example, “h colors”). The graphics control help screen then gives some information on that option. You may select several options, or select no options. Either way, a plot is produced. More information on how to select options and how to save and use options is described in detail later in this paper.

After the options are selected, then you simply hit “r” and return to produce your plot. After the plot is produced, more options are offered:

```
To save these graphics keyword settings, enter s
and a space followed by a file name;
```

OR

```
To save the plot as a color postscript file, enter cp
and a space followed by a file name;
```

OR

```
To save the plot as a black and white postscript file, enter bw
and a space followed by a file name;
```

OTHERWISE, press enter.

The user may wish to print the plot in a postscript, selecting either the color or the black/white option. Simply pressing enter will not produce a plot for printing. After pressing “enter”, an option of continuing with a different trace.dat file is offered. A user with a trace.dat file with different conditions may select this option, but in general this is ignored.

Pressing “enter” brings the user back to Menu 1. Here the user may select a new plot or simply exit fvs by pressing “0” and enter.

Most of the different options are produced from Menu 4. From there you may plot the RMS, bias, correlation, covariance, standard deviation, or simply plot the observations against the one or more models.

The above verification items apply to scalar verification. Most of the verification items are scalars. However, for vector wind verification (VWND), some of the items are slightly different.

To plot vector wind verification performance measures, the following must be included in the control file:

```
1 10 0 2 1 1 1 0/0/1/0
18 0 0 VL1L2 0
2 10 0 2 1 1 1 0/0/1/0
18 0 0 VL1L2 0
1 12 0 2 1 1 1 0/0/1/0
18 0 0 VWND 0
2 12 0 2 1 1 1 0/0/1/0
```

18 0 0 VWND 0

Note that not only does the variable need to be VWND, but the statistic type is also changed. Instead of SL1L2, scalar statistics being plotted, the statistic type is now VL1L2, vector statistics.

Most of the plotting items are the same, but Menu 4 is slightly different. Here is what it looks like:

```
*****
* VL1L2 -- L1L2 norm statistics for vectors *
* * * * *
* Computation Code Action *
* * * * *
* 9002 Trace of data field values averaged over all pts*
* -n Return raw data for field n (undivided sum) *
* -1 Contributing record count at all trace points *
* 1 Return data from field 1 = N, the data count *
* 2 Return data from field 2 = U forecast mean *
* 3 Return data from field 3 = V forecast mean *
* 4 Return data from field 4 = U observed mean *
* 5 Return data from field 5 = V observed mean *
* 6 Return data from field 6 = MEAN [Uf*Uo+Vf*Vo] *
* 7 Return data from field 7 = MEAN [Uf**2+Vf**2] *
* 8 Return data from field 8 = MEAN [Uo**2+Vo**2] *
* * * * *
* 1201 Compute RMS speed BIAS (V[S]AL1L2, VL1L2) *
* 1202 Compute vector RMS error (V[S]AL1L2, VL1L2) *
* 1203 Compute vector Correlation (V[S]AL1L2, VL1L2) *
* 1204 Compute vector Covariance (V[S]AL1L2, VL1L2) *
* 1205 Compute vector Frcst Variance (V[S]AL1L2, VL1L2)*
* 1206 Compute vector Obs Variance (V[S]AL1L2, VL1L2) *
* 1207 Compute vector Frcst Std Dev (V[S]AL1L2, VL1L2) *
* 1208 Compute vector Obs Std Dev (V[S]AL1L2, VL1L2) *
* 1209 Compute U comp BIAS (V[S]AL1L2, VL1L2) *
* 1210 Compute V comp BIAS (V[S]AL1L2, VL1L2) *
* 1211 Compute vector BIAS magnitude (V[S]AL1L2, VL1L2)*
* 1212 Compute vector Error Std Dev (V[S]AL1L2, VL1L2) *
* 1213 Compute vector BIAS**2 *
* 1214 Compute -2 * (vector Covariance) *
* 1215 Compute vector Error Variance *
* 1216 Compute vector Mean Squared Error *
*****
```

Select computational code from table above. Please
enter one computational code(s).

Many of the above items are similar to the SL1L2 menu, but there are a few differences, since there are more elements in a wind VSDB data record. Various error statistics can be displayed. The vector bias can be plotted, or just the U or V bias. Most of the statistics shown above are the same as in the scalar statistics, but since they are vectors they are calculated slightly differently. Other than the changes to the control file and this menu, very few other differences exist between scalar verification and vector verification.

Upper-air profiles

Here is an example of a control file for an upper-air profile:

```

1 14 1 2 8 1 4 0/0/1/0
18 0 0 P1000 0
18 0 0 P850 0
18 0 0 P700 0
18 0 0 P500 0
18 0 0 P400 0
18 0 0 P300 0
18 0 0 P250 0
18 0 0 P200 0
2 14 1 2 8 1 4 0/0/1/0
18 0 0 P1000 0
18 0 0 P850 0
18 0 0 P700 0
18 0 0 P500 0
18 0 0 P400 0
18 0 0 P300 0
18 0 0 P250 0
18 0 0 P200 0
1 1 0 2 1 1 4 0/0/1/0
18 0 0 V01 0
2 1 0 2 1 1 4 0/0/1/0
18 0 0 V01 0
1 2 0 2 1 1 4 0/0/1/0
18 0 0 NAM/218 0
2 2 0 2 1 1 4 0/0/1/0
18 0 0 GFS/212 0
1 3 0 2 1 1 4 0/0/1/0
9 06 0 0 0
2 3 0 2 1 1 4 0/0/1/0
9 06 0 0 0
1 5 0 2 1 1 4 0/0/1/0
15 0 0 200611010000 200612100000
2 5 0 2 1 1 4 0/0/1/0
15 0 0 200611010000 200612100000
1 8 0 2 1 1 4 0/0/1/0
18 0 0 ADPUPA 0
2 8 0 2 1 1 4 0/0/1/0
18 0 0 ADPUPA 0
1 9 0 2 1 1 4 0/0/1/0
18 0 0 G236 0
2 9 0 2 1 1 4 0/0/1/0
18 0 0 G236 0
1 10 0 2 1 1 4 0/0/1/0
18 0 0 SL1L2 0
2 10 0 2 1 1 4 0/0/1/0
18 0 0 SL1L2 0
1 12 0 2 1 1 4 0/0/1/0
18 0 0 T 0
2 12 0 2 1 1 4 0/0/1/0
18 0 0 T 0

```

The main difference in this control file compared to the other one is the listing of levels:

```

1 14 1 2 8 1 4 0/0/1/0
18 0 0 P1000 0
18 0 0 P850 0
18 0 0 P700 0
18 0 0 P500 0
18 0 0 P400 0
18 0 0 P300 0
18 0 0 P250 0

```

```

18 0 0 P200 0
2 14 1 2 8 1 4 0/0/1/0
18 0 0 P1000 0
18 0 0 P850 0
18 0 0 P700 0
18 0 0 P500 0
18 0 0 P400 0
18 0 0 P300 0
18 0 0 P250 0
18 0 0 P200 0

```

Here note that the levels in this case is the independent variable, the others are the dependent variables. Also note the 5th element in each header line here represents the number of levels to follow (in this case 8). In the example above, every one of the 5th elements was 1. One may also have any number, up to the prescribed limits, of elements in each section (such as the verification areas, or verification dataset types), either as independent or dependent variables. If it is a dependent variable, the verification is averaged over all of the elements in the list.

Here are the limits for each section:

- The max number of conditionally determined independent variable entries is 64.
- The max number of independent variable conditions is 8.
- The max number of dependent variable conditions is 16.
- The max number of elements in the dependent variable search list is 32.
- The max number of traces is 24.

Also note that the 7th element is different. Before that number was a “1”, representing that the plot is a time series. This time it is a “4”, denoting that we have a “categorically binned data plot”. That means that since the levels are the dependent variable, the verification is being done for each level, which is considered a “categorical bin”, rather than an average over the entire profile.

After running the “fvs s” searcher, the options described above in the plotting program are exactly the same. One exception is that when you get to Menu 5 (described in the previous section), it is best to type in the following:

```
revers=y
```

This will reverse the orientation of the x-axis and the y-axis, so that the result is a profile, with the pressure levels starting at 1000 at the bottom of the axis, and going up to 200, which is what one desires to see when examining a profile. However, take caution when doing the other settings, because the pressure levels is still considered the x-axis when you do your settings, even with the revers=y option set.

With the profile, you can plot out the very same items described above in the time series section. The difference is that time is no longer an independent variable. The vertical plot that is seen is an average over the time selected in the plot, as denoted in the control file.

Finally, as in the time series, slight differences exist when plotting out vector verifications, but they are the same differences as noted above for time series.

Diurnal cycle

This type of plot is similar to a time series, but this time the dependent variable is the forecast time. With this plot one can view the verifications throughout the forecast. Like the upper-air profile described in the last section, the final plot is an average of the results for the time period selected in the control file.

To make this plot, one simply has to add the following to a control file:

```

1 2 1 2 29 1 4 0/0/1/0
9 00 0 0 0
9 03 0 0 0
9 06 0 0 0
9 09 0 0 0
9 12 0 0 0
9 15 0 0 0
9 18 0 0 0
9 21 0 0 0
9 24 0 0 0
9 27 0 0 0
9 30 0 0 0
9 33 0 0 0
9 36 0 0 0
9 39 0 0 0
9 42 0 0 0
9 45 0 0 0
9 48 0 0 0
9 51 0 0 0
9 54 0 0 0
9 57 0 0 0
9 60 0 0 0
9 63 0 0 0
9 66 0 0 0
9 69 0 0 0
9 72 0 0 0
9 75 0 0 0
9 78 0 0 0
9 81 0 0 0
9 84 0 0 0
2 2 1 2 29 1 4 0/0/1/0
9 00 0 0 0
9 03 0 0 0
9 06 0 0 0
9 09 0 0 0
9 12 0 0 0
9 15 0 0 0
9 18 0 0 0
9 21 0 0 0
9 24 0 0 0
9 27 0 0 0
9 30 0 0 0
9 33 0 0 0
9 36 0 0 0
9 39 0 0 0

```

```

9 42 0 0 0
9 45 0 0 0
9 48 0 0 0
9 51 0 0 0
9 54 0 0 0
9 57 0 0 0
9 60 0 0 0
9 63 0 0 0
9 66 0 0 0
9 69 0 0 0
9 72 0 0 0
9 75 0 0 0
9 78 0 0 0
9 81 0 0 0
9 84 0 0 0

```

Note that this is every output time in a 0-84 hour forecast by three hour increments. For a shorter time period, simply change the 5th element of the header to a smaller number and remove any forecast hours that are not desired to plotting. Note that the 7th element of the header is a 4, denoting that this is another type of categorically binned data plot.

As before, the “fvs p” option will present the same options as before, giving the user the same options as the time series and/or the upper-air profile to plot as statistics. Also, this option is available as well for both scalar variables and vector variables.

Tropical Cyclone data

Tropical cyclone data in VSDB format can now be displayed with FVS. The VSDB record for tropical cyclones are different than those for the above data sets, but their treatment in FVS is similar. This particular datatype is known as the TTRK statistic type, and a data record contains different information than the SL1L2 datatype. Here are the data fields in a TTRK data record:

DATA Field	Contents
1	Number of contributing storms, usually 1
2	Latitude (-90 -> +90) of observed position
3	Longitude (-180 -> +180) of observed position
4	Direction (0 -> 360) from north at observed position to vector pointing along great circle arc toward forecast position
5	Distance in km along great circle arc from observed to forecast position
6	Distance (5) squared
7	X-component (km) of vector position error in standard meteorological coordinates
8	Y-component (km) of vector position error in standard meteorological coordinates; direction(4)=arctan (x/y) and distance squared(6)=x**2+y**2.
9--13	SL1L2 norms for central pressure (mb)
14--18	SL1L2 norms for maximum wind speed (m/s)
	-End of MTRK Record-
19--23	SL1L2 norms for 850 vorticity maximum (/s)

A trace.ctl file for tropical cyclones would look like this:


```

1 5 1 2 1 1 1 0/0/0/0
15 0 0 200508021800 200508271800
2 5 1 2 1 1 1 0/0/0/0
15 0 0 200508021800 200508271800
1 1 0 2 1 1 1 0/0/0/0
18 0 0 V01 0
2 1 0 2 1 1 1 0/0/0/0
18 0 0 V01 0
1 2 0 2 1 1 1 0/0/0/0
18 0 0 ETA 0
2 2 0 2 1 1 1 0/0/0/0
18 0 0 GFDL 0
1 3 0 2 1 1 1 0/0/0/0
9 24 0 0 0
2 3 0 2 1 1 1 0/0/0/0
9 24 0 0 0
1 8 0 2 1 1 1 0/0/0/0
18 0 0 TPC 0
2 8 0 2 1 1 1 0/0/0/0
18 0 0 TPC 0
1 9 0 2 1 1 1 0/0/0/0
18 0 0 AL 0
2 9 0 2 1 1 1 0/0/0/0
18 0 0 AL 0
1 10 0 2 1 1 1 0/0/0/0
18 0 0 TTRK 0
2 10 0 2 1 1 1 0/0/0/0
18 0 0 TTRK 0
1 14 0 2 1 1 1 0/0/0/0
18 0 0 X 0
2 14 0 2 1 1 1 0/0/0/0
18 0 0 X 0
1 12 0 2 1 1 1 0/0/0/0
18 0 0 082005 0
2 12 0 2 1 1 1 0/0/0/0
18 0 0 082005 0

```

This is a time series that is very similar to what was shown before. The difference is that this points to time series data. Here is what to note:

- a) For the data type, TPC is used rather than, say ADPUPA or ONLYSF or another commonly-used BUFR data type.
- b) For the output grid, rather than using one of the G grids, AL is used.
- c) Instead of the SL1L2 or FHO datatypes, the TTRK datatype is used.
- d) Instead of the variable to be plotted, such as Z, T, RH, etc, a number is used. In this example, it is 082005. That is a number that denotes the time that the tropical cyclone formed. To get the statistics over more than one cyclone, the user should put in a wild card here by simply typing a "0" where the 082005 is (or type "wild" when running this option in "fvs u").
- e) Rather than have a level, such as P850, only an X is required.

Running fvs on this control file gives an entirely different Menu 4:

```
*****
```

```

* MTRK and TTRK - Storm Track Error Statistics *
* *
* Computation Code Action *
* *
* 9002 Trace of data field values averaged over all pts*
* -n Return raw data for field n (undivided sum) *
* -1 Contributing record count at all trace points *
* 1 Return data from field 1 = N, the data count *
* 2 Return data from field 2 = mean obs'd latitude *
* 3 Return data from field 3 = mean obs'd longitude *
* 4 Return data from field 4 = avg angle from N. *
* 0-360; do not display*
* 5 Return data from field 5 = mean dist error *
* 6 Return data from field 6 = mean dist error ** 2 *
* 7 Return data from field 7 = mean x position error*
* 8 Return data from field 8 = mean y position error*
* 9-13 L1L2 norms (f,o,fo,ff,oo) for central MSLP *
* 14-18 TTRK L1L2 norms (f,o,fo,ff,oo) for max wind spd *
* 19-23 TTRK L1L2 norms (f,o,fo,ff,oo) for 850 Vort Max *
* *
* x1101 Compute BIAS from L1L2 norms *
* x1102 Compute RMS error from L1L2 norms *
* x1103 Compute Correlation from L1L2 norms *
* x1104 Compute Covariance from L1L2 norms *
* x1105 Compute Forecast Variance from L1L2 norms *
* x1106 Compute Obs Variance from L1L2 norms *
* x1107 Compute Forecast Std Dev from L1L2 norms *
* x1108 Compute Obs Std Dev from L1L2 norms *
* x1109 Compute Error Std Dev from L1L2 norms *
* x1110 Compute BIAS + Error Std Dev from L1L2 norms *
* x1111 Compute BIAS - Error Std Dev from L1L2 norms *
* x1112 Compute BIAS**2 *
* x1113 Compute -2*Covariance *
* x1114 Compute Error Variance *
* x1115 Compute Mean Squared error *
* *
* where x = 10, 11, 12, or 13 as indicated below: *
* *
* 101101 - 101115 L1L2 computations for TRK distance error. *
* 111101 - 111115 L1L2 computations for TRK PMSL error. *
* 121101 - 121115 L1L2 computations for TRK Max Wind speed error. *
* 131101 - 131115 L1L2 computations for TRK 850 Vort error. *
* *
* 1801 Average directional error for TRK stats *
*****

```

This particular menu is specific to the verification of tropical cyclones. There are similarities to the Menu 4 for the regular time series, but in this case there are some differences. The upper portion allows the user to print out cyclone track latitude and longitudes, position errors, as well as ob vs. model for items for central sea level pressure, max wind speed, and 850 mb vorticity maxima, all important cyclone characteristics.

The user can also plot the same statistics for each variable, such as bias, RMS errors, covariances, etc, for the distance, central pressure, max wind, and 850 mb vorticity maxima as the state variables. Depending on which the user prefers to plot, a two-digit number is added in front of the usual statistic.

FHO Statistics

In addition to all the above statistics (called the SL1L2 statistics), such as root mean square, covariances, standard deviations, and the like, one can plot another category of statistics, called the FHO statistics. FHO stands for Forecast, Hit, Observation. These statistics provide a measure of how often we get “hits”, that is the forecast comes close to matching the observation. The statistics are plotted against thresholds (for example, how many times a forecast of 10 degrees or greater is correct, how many times a forecast of 15 degrees or greater is correct, etc.).

FHO requires a different VSDB format than the SL1L2 statistics. Their creation is described in the NCEP Verification System Documentation paper. A VSDB file can contain both SL1L2 partial sums and FHO partial sums, and the FVS codes will compute and plot the statistics for both.

The trace.ctl file is not all that different than the time series control file. There are only two differences. First, the control file must contain the list of thresholds to verify. Those thresholds become the independent variable. Add this to the file:

```
1 11 1 2 7 1 4 0/0/0/0
4 0 0 0 0
4 5 0 0 0
4 10 0 0 0
4 15 0 0 0
4 20 0 0 0
4 25 0 0 0
4 30 0 0 0
2 11 1 2 7 1 4 0/0/0/0
4 0 0 0 0
4 5 0 0 0
4 10 0 0 0
4 15 0 0 0
4 20 0 0 0
4 25 0 0 0
4 30 0 0 0
```

This is showing that there are 7 thresholds we are verifying: 0, 5, 10, 15, 20, 25, and 30 (in this example the control file is verifying surface dew point). The 2nd element, 11, is the “fvs u” selection number for thresholds (see above). The number 11 represents plotting thresholds for data.

The other change is the statistic type:

```
1 10 0 2 1 1 4 0/0/0/0
18 0 0 FHO 0
2 10 0 2 1 1 4 0/0/0/0
18 0 0 FHO 0
```

Note that this file has “FHO” in the place where there was “SL1L2” previously. An individual plot in a control file can have either FHO verification or SL1L2 verification, but not

both. A different plot or a different control file must be used in order to do both types of verification with one data set.

Almost all the fvs p menus are the same as before, with the exception of Menu 4, which now is different, as shown below.

```
*****
* FHO -- Forecast, hit, observation rate statistics *
* *
* Computation Code Action *
* *
* 9002 Trace of data field values averaged over all pts*
* -n Return raw data for field n (undivided sum) *
* -1 Contributing record count at all trace points *
* 1 Return data from field 1 = N, the data count *
* 2 Return data from field 2 = F, fraction forecast *
* 3 Return data from field 3 = H, fraction correct *
* 4 Return data from field 4 = O, fraction observed *
* *
* 1001 Compute BIAS from FHO values *
* 1002 Compute Equitable Threat Score from FHO values *
* 1003 Compute Critical Success Index from FHO values *
* 1004 Compute False Alarm Ratio from FHO values *
* 1005 Compute Postagreement from FHO values *
* 1006 Compute Prefigurance from FHO values *
* 1007 Compute Probability of Detection from FHO values*
* 1008 Compute Threat Score from FHO values *
* 1009 Cumulative Forecast Prcp Vol % vs Thrshld *
* 1010 Cumulative Correctly Frcst Prcp Vol % vs Thrshld*
* 1011 Cumulative Observed Prcp Vol % vs Thrshld *
* 1012 Forecast Prcp Vol % of Ttl vs Thrshld *
* 1013 Correctly Forecast Prcp Vol % of Ttl vs Thrshld *
* 1014 Observed Prcp Vol % of Ttl vs Thrshld *
* 1015 Forecast Prcp # % of Ttl # vs Thrshld *
* 1016 Correctly Forecast Prcp # % of Ttl # vs Thrshld *
* 1017 Observed Prcp # % of Ttl # vs Thrshld *
* 1018 Forecast # % of Ttl <= Thrshld vs Thrshld *
* 1019 Correctly Frcst # % of Ttl <= Thrshld vs Thrshld*
* 1020 Observed # % of Ttl <= Thrshld # vs Thrshld *
* 1021 Compute False Alarm Rate (f-h)/(N-o) *
* 1022 Compute Conditional Miss Rate (o-h)/(N-f) *
* *
* 1023 Compute Heidke Skill Score *
* 1024 Compute Peirce-Hanssen-Kuipers Skill Score *
* 1025 Compute ODDS RATIO (OR) *
* 1026 Compute log of ODDS RATIO *
* 1027 Compute standard error of log of ODDS RATIO *
* 1028 Compute ODDS RATIO Skill Score *
* 1029 Compute ODDS RATIO Skill Score Cubed *
* 1030 Compute log (ODDS RATIO) / STD ERROR of same *
* 1031 Area between ODDS RATIO ROC & chance ROC curves *
* 1032 ROC curve parametric in OR for time series *
* 1033 Odds ratio Parameterized ROC area (OPRA) score *
* 1034 Frcst accuracy rate (hits and correct nonevents)*
* 1035 Extreme Dependency Score (EDS) *
* *
* Any of the preceding codes may be computed from bias adjusted FHO *
* values. To do this, precede the 4-digit code number with one of *
* the following bias adjustment specification prefixes: *
* *
* 101 Fedor Mesinger's dH/dF bias adjustment *
* 102 Conservation of Odds Ratio bias adjustment *
```

```

*
* Example 1: -1020003 displays the bias adjusted hit count
* Example 2: 1021008 displays the bias adjusted threat score
*
* Any of the preceding 10xx series that are N dependent may be
* computed using F+O-H from the Mth preceding threshold. The first M
* thresholds use the original value of N, the total number. Thres-
* hold must be the independent variable. To do this, precede the
* 10xx code with the following prefix:
*
*      20M The F+O-H value from i-M threshold is used, M>0 and M<10
*****
Select computational code from table above. Please
enter one computational code(s).

```

This menu presents a whole new list of different items to plot that were not made available with the SL1L2 verification. Items like equitable threat scores, false alarm ratios, probability of detection, and several skill scores can be calculated from this set. As in the case of the upper-air profile and the diurnal cycle, the result is an average calculated over the entire time period set in the control file.

Ensemble statistics

Ensemble verification has its own set of statistics. Each deterministic forecast can be verified using one of the methods described above, but statistics have also been created to determine the quality of the entire ensemble itself.

Here is an example of a trace.ctl file for ensembles:

```

1 2 0 2 1 1 1 2/2/0/0
18 0 0 SREF/21 0
2 2 0 2 1 1 1 2/2/0/0
18 0 0 SREF/CTL 0
3 2 0 2 1 1 1 2/2/0/0
18 0 0 SREF/ETA 0
4 2 0 2 1 1 1 2/2/0/0
18 0 0 SREF/RSM 0
5 2 0 2 1 1 1 2/2/0/0
18 0 0 SREF/WRF 0
1 3 0 2 1 1 1 2/2/0/0
9 51 0 0 0
2 3 0 2 1 1 1 2/2/0/0
9 51 0 0 0
3 3 0 2 1 1 1 2/2/0/0
9 51 0 0 0
4 3 0 2 1 1 1 2/2/0/0
9 51 0 0 0
5 3 0 2 1 1 1 2/2/0/0
9 51 0 0 0
1 14 0 2 1 1 1 2/2/0/0
18 0 0 SFC 0
2 14 0 2 1 1 1 2/2/0/0
18 0 0 SFC 0
3 14 0 2 1 1 1 2/2/0/0
18 0 0 SFC 0
4 14 0 2 1 1 1 2/2/0/0
18 0 0 SFC 0

```

```

5 14 0 2 1 1 1 2/2/0/0
18 0 0 SFC 0
1 8 0 2 1 1 1 2/2/0/0
18 0 0 ONLYSF 0
2 8 0 2 1 1 1 2/2/0/0
18 0 0 ONLYSF 0
3 8 0 2 1 1 1 2/2/0/0
18 0 0 ONLYSF 0
4 8 0 2 1 1 1 2/2/0/0
18 0 0 ONLYSF 0
5 8 0 2 1 1 1 2/2/0/0
18 0 0 ONLYSF 0
1 9 0 2 1 1 1 2/2/0/0
18 0 0 G236 0
2 9 0 2 1 1 1 2/2/0/0
18 0 0 G236 0
3 9 0 2 1 1 1 2/2/0/0
18 0 0 G236 0
4 9 0 2 1 1 1 2/2/0/0
18 0 0 G236 0
5 9 0 2 1 1 1 2/2/0/0
18 0 0 G236 0
1 10 0 2 1 1 1 2/2/0/0
18 0 0 ESL1L2/5 0
2 10 0 2 1 1 1 2/2/0/0
18 0 0 ESL1L2/5 0
3 10 0 2 1 1 1 2/2/0/0
18 0 0 ESL1L2/5 0
4 10 0 2 1 1 1 2/2/0/0
18 0 0 ESL1L2/5 0
5 10 0 2 1 1 1 2/2/0/0
18 0 0 ESL1L2/5 0
1 12 0 2 1 1 1 2/2/0/0
18 0 0 T 0
2 12 0 2 1 1 1 2/2/0/0
18 0 0 T 0
3 12 0 2 1 1 1 2/2/0/0
18 0 0 T 0
4 12 0 2 1 1 1 2/2/0/0
18 0 0 T 0
5 12 0 2 1 1 1 2/2/0/0
18 0 0 T 0
1 5 1 2 1 1 1 2/2/0/0
15 0 0 200701010000 200702152359
2 5 1 2 1 1 1 2/2/0/0
15 0 0 200701010000 200702152359
3 5 1 2 1 1 1 2/2/0/0
15 0 0 200701010000 200702152359
4 5 1 2 1 1 1 2/2/0/0
15 0 0 200701010000 200702152359
5 5 1 2 1 1 1 2/2/0/0
15 0 0 200701010000 200702152359

```

This particular example plots verification for the 21-member ensemble vs. the ensemble control as well as the Eta, RSM, and WRF. This particular example is the 51-hr 2-m temperature, but it could be any variable. The relevant portion of the control file is below:

```

1 10 0 2 1 1 1 2/2/0/0
18 0 0 ESL1L2/5 0
2 10 0 2 1 1 1 2/2/0/0
18 0 0 ESL1L2/5 0
3 10 0 2 1 1 1 2/2/0/0
18 0 0 ESL1L2/5 0

```

```

4 10 0 2 1 1 1 2/2/0/0
18 0 0 ESL1L2/5 0
5 10 0 2 1 1 1 2/2/0/0
18 0 0 ESL1L2/5 0

```

This datatype, ESL1L2/5, is similar to the SL1L2 that has been described in the single forecasts in the sections above. The covariances, RMS errors, bias errors, etc, are all available with this datatype, and the VSDB record for this type is the same as the others described above. Two more different datatypes will be introduced in this section, RHET and RHNT. These datatypes have different VSDB records, and it is assumed in this document that the user has software to create these VSDB records.

For the ESL1L2 ensemble mean scalar statistics, the Menu 4 choices are similar to the Menu 4 from the SL1L2 with a few additions:

```

*****
* ESL1L2 -- Ensemble Mean Scalar L1L2 norm statistics *
* * * * *
* Computation Code Action *
* * * * *
* 9002 Trace of data field values averaged over all pts*
* -n Return raw data for field n (undivided sum) *
* -1 Contributing record count at all trace points *
* 1 Return data from field 1 = N, the data count *
* 2 Return data from field 2 = forecast mean *
* 3 Return data from field 3 = observation mean *
* 4 Return data from field 4 = F * O mean *
* 5 Return data from field 5 = F ** 2 mean *
* 6 Return data from field 6 = O ** 2 mean *
* 7 Return data from field 7 = Variance wrt ens mean*
* 8 Data field 8 = Verification outlier fraction *
* * * * *
* 1101 Compute BIAS (ES[S]AL1L2, ESL1L2) *
* 1102 Compute RMS error (ES[S]AL1L2, ESL1L2) *
* 1103 Compute Correlation (ES[S]AL1L2, ESL1L2) *
* 1104 Compute Covariance (ES[S]AL1L2,ESL1L2) *
* 1105 Compute Forecast Variance (ES[S]AL1L2, ESL1L2) *
* 1106 Compute Obs Variance (ES[S]AL1L2, ESL1L2) *
* 1107 Compute Forecast Std Dev (ES[S]AL1L2, ESL1L2) *
* 1108 Compute Obs Std Dev (ES[S]AL1L2, ESL1L2) *
* 1109 Compute Error Std Dev (ESAL1L2, ESL1L2) *
* 1110 Compute BIAS + Error Std Dev (ESAL1L2, ESL1L2) *
* 1111 Compute BIAS - Error Std Dev (ESAL1L2, ESL1L2) *
* 1112 Compute BIAS**2 *
* 1113 Compute -2*Covariance *
* 1114 Compute Error Variance *
* 1115 Compute Mean Squared error *
* 1116 Scalar ensemble statistical consistency index *
*****

```

Some of the differences here include option 7, the variance with respect to the ensemble mean, option 8, the verification outlier fraction, and option 1116, the ensemble consistency index.

If the datatype is changed to RHET, this is the Menu 4 that appears:

```

*****

```

```

*
* RHET -- Probabilities that ensemble members encompass observed or
*         analyzed data
*
* Computation Code          Action
*
*   9002                    Trace of data field values averaged over all pts*
*   -n                      Return raw data for field n (undivided sum)
*   -1                      Contributing record count at all trace points
*   1                        Return data field 1 = N, the data count
*   2                        Data field 2 = probability of analysis is less
*                             than lowest ensemble member
*   3                        Data field 3 = probability of analysis between
*                             lowest and next lowest members
*   ...
*   Nm + 1                  Data field Nm+1 = probability of analysis
*                             between highest and next
*                             highest ensemble members
*                             Nm is the number of members
*
*   1600                    Display as Ranked Histogram Analysis Diagram
*   1601                    Probability analysis is lower than lowest
*                             ensemble member
*   1602                    Probability analysis is within the range of
*                             ensemble members
*   1603                    Probability analysis is higher than highest
*                             ensemble member
*   1604                    Probability analysis is outside range of the
*                             ensemble members
*   1605                    Display 1601--1604 as 4-point trace
*   1606                    Missing rate error (MRE)
*****

```

These are the probabilities of ensemble members encompassing observed or analyzed data. The ranked histogram provides a convenient way to show such information. The user can also compare the probabilities of the analysis being within the ensemble members.

The data record for a RHET datatype is different than the datatype for a SL1L2 record. It consists of the following.

DATA Field	Contents
1	Number of contributing verification events
2	probability analysis less than lowest ensemble member
3	probability analysis between lowest and next lowest
...	
Nm+1	probability analysis between next highest and highest
residual =	probability analysis greater than highest ensemble member

If the datatype is changed to RHNT, this is the Menu 5 that appears:

```

*****
*
* RHNT -- probabilities of ensemble members being nearest observed or
*         analyzed data
*
* Computation Code          Action
*
*   9002                    Trace of data field values averaged over all pts*
*   -n                      Return raw data for field n (undivided sum)
*   -1                      Contributing record count at all trace points
*   1                        Return data field 1 = N, the data count

```



```

*          2          Data field 2 = probability of analysis closest *
*                                     to ensemble member # 1 *
*          3          Data field 3 = probability of analysis closest *
*                                     to ensemble member # 2 *
*          ...
*          Nm         Data field Nm = probability of analysis closest *
*                                     to ensemble member # Nm - 1, *
*                                     Nm is the number of members *
*
*          1600       Display as Ranked Histogram Analysis Diagram *
*****

```

This particular datatype calculates the probability of the analysis being closest to each ensemble member. Again, this particular datatype has a different data record that looks like the following:

DATA Field	Contents
1	Number of contributing verification events
2	probability analysis closest to ensemble member # 1
3	probability analysis closest to ensemble member # 2
...	
Nm	probability analysis closest to ensemble member # Nm-1
residual =	probability analysis closest to ensemble member # Nm

For both RHNT and RHET the user is offered the choice to display the information summarized in a Ranked Histogram. For RHET, the plot is also known as a Talagrand Diagram. For RHNT, the plot is known as an Equally Likely Diagram.

Scatterplots

There are several items that one can plot other than what is listed above. For example, if we go back to Menu 2:

```

Select plot display type by number:

    0 - Print statistics for traces in plot defined above.

    1 - Graphical display of plot as defined above.

    2 - Scatter plot (Requires 2 series of computational
        results, either from the same trace or two
        different traces.)

    3 - Old-style histogram (requires exactly one trace)

    4 - Display scatter plot data as line plots

    5 - Display ROC curves (select comp code 1032 below)

Enter choice:

```

Selection 2 here produces a scatter plot. Here one can see how two items, such as an

observation set and a forecast, are correlated. Enter 2, and the same Menu 3 as above appears. In this case it is generally best to enter 2 again:

```
2 - Explicitly enter trace numbers and corresponding
    computational codes
```

Then once the trace numbers are entered, the Menu 4 items are entered (2 is the forecast mean and 3 is the observation mean, though any two statistics can be entered). Then once the plot is created, a scatterplot is shown, with the first variable entered on the x-axis and the second on the y-axis. Individual obs-model combinations are plotted, with the $y=x$ line plotted as well. The regression line of the scatterplot is also plotted so one can compare the slope of that line with the $y=x$ line ($y=x$ being perfect correlation between obs and model forecasts). The correlation is printed at the top of the graph.

Bar graphs

When a bar graph is desired, rather than a line graph, very few changes to the process are required. The same selections in fvs p are done. When the last menu, Menu 5, appears, then the signal to get a bar graph is to set up the option for the color of the bars. That signal is indicated with the HSTCLR option. For example:

```
HSTCLR=1/2
```

can be typed before the graph is plotted. By doing this, you not only signify that a bar graph is desired, but the colors are also set so each bar can be distinguished from the others. Bar thickness is automatically determined based on the number of bars. In the meantime, if the “r” is typed after this is entered into the options, then a bar graph will be created.

Significance testing

Significance testing is also known as hypothesis testing. Such testing can be done to determine if there is a statistically significant difference between the paired data of two traces. The null hypothesis is that there is zero difference between the two traces. The alternative hypothesis is posed diagnostically. The reject or accept null hypothesis decision is given for several p-values, which are probabilities of falsely rejecting the null hypothesis.

FVS performs significance testing on time series. One thing that is very important regarding performing significance testing is that the number of points for each trace must be equal. That does not have to be the case for plotting statistics in general, though.

To ensure that the number of points is equal, consistency checking must be performed. The

easiest way to do this is to go into the control file, and change, for example, the following:

```
1 5 1 2 1 1 1 0/0/1/0
15 0 0 200611010000 200611100000
2 5 1 2 1 1 1 0/0/1/0
15 0 0 200611010000 200611100000
```

Change the above to:

```
1 5 1 2 1 1 1 2/2/1/0
15 0 0 200611010000 200611100000
2 5 1 2 1 1 1 2/2/1/0
15 0 0 200611010000 200611100000
```

The only change that appears in the above two sections of the control file is in the last set of four numbers separated by a slash. Note that the first in this sequence has been changed from a 0 to a 2. Changing it to a 2 ensures that consistency checking is performed so that the end result is that the record IDs and counts must match. A zero in that space performs no consistency checking. The 2 in the second space indicates that each point is consistent through all traces, but variation from point to point is allowed. This change must be done throughout the entire trace.ctf file.

After this is done, the searching and plotter is done as usual. When Menu 2 appears, the following option should be chosen:

```
6 - Explicitly enter two trace numbers and one computational
    code (if trace numbers are identical, enter two different
    computational codes) to compute statistical significance of
    the differences of the paired values of the two traces
```

Then the user is asked to enter the two trace numbers and the computational code. At the final menu, running the program gives no plot. Instead, the following appears:

```
See file fvs_sig_test1.txt for output.
```

Exiting the program and looking at the output text file gives the user the results of the significance test:

```
Null Hypothesis (tracel = trace2):
  The mean of the differences of the paired values from the
  two traces is zero.

Alternative Hypothesis (tracel < trace2):
  The mean of the paired differences is less than zero.

Gaussian probability of wrongly rejecting null hypothesis:  0.022861

Test Level  Decision (one-sided Gaussian)
  0.100      REJECT null hypothesis
```

```

0.050      REJECT null hypothesis
0.010      ACCEPT null hypothesis
0.005      ACCEPT null hypothesis
0.001      ACCEPT null hypothesis

```

```

ZRAW = -1.91333E+00  ZTST = -1.99795E+00  N =      58  Lag1 Corr = -0.043252

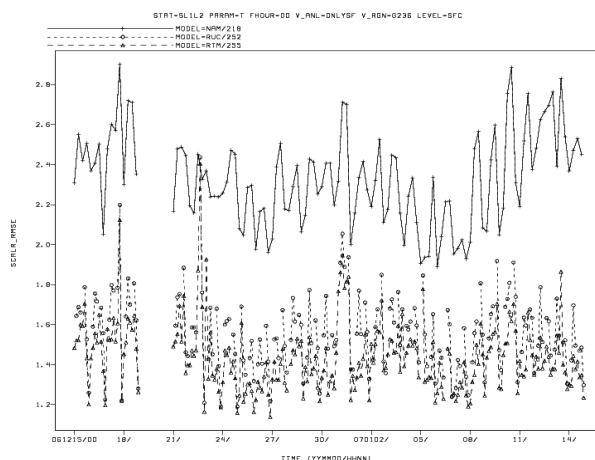
```

From here it is up to the user to determine whether or not to reject the null hypothesis, defined in the output as being the difference of the two traces is zero. This can give some sort of idea as to the comparative skill or the accuracy of the model forecasts.

Changing appearance of plots

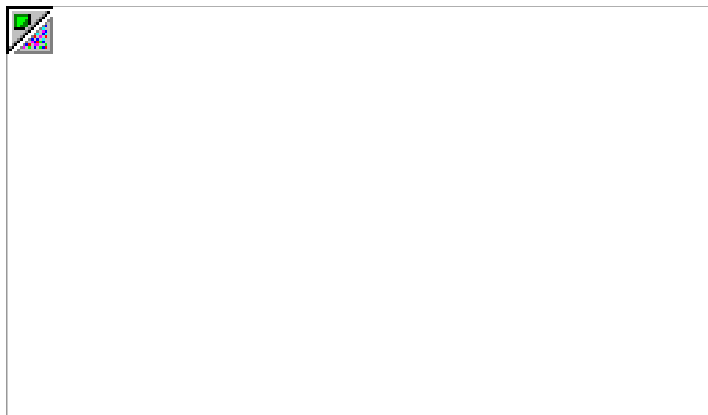
Several examples have been shown on how one can adjust their plots so they are nice-looking and easy to read. Menu 5 above gave the options on how one can adjust their plots. The user can either do so on the fly or save their plot adjustment codes in the “settings” file. The settings file can then be invoked on future applications of the plotting routines. Below are a few examples of what can be set in order to get a nice plot.

If no settings are selected, a particular plot might look like this:



This is kind of plain looking, with computer-generated graphics and labels. The header in this example, while giving the details of what is contained in the plot, it uses FVS lingo and is not clearly understood what is contained in the plots. It is not ready for a presentation.

On the other hand, here is an example of a nicer-looking plot:



This particular plot is much clearer. It makes the use of color, thicker lines, larger symbols, and a clearer plot title written in plain English. This is the type of plot that FVS is capable of creating if one uses one or more of the settings described below.

1) LINE

The user can set the color, type of line, the width, and the marker frequency of the plot line. The command looks like this:

```
colr1;...;colrn/typel;...;typen/width1;...;widthn/lab11;...;labln
```

The way this reads is that the color is set first. If there are n lines to be plotted, then color of the line is set for each line, with each color number separated by semicolons (;). Each attribute is set this way. Between each attribute, a slash (/) is used.

Typing “fvs h colors” gives a description of all the colors and their number codes. There are 32 colors that the user can use (for brevity they are not listed here).

Next the type of line is selected. There are ten line types:

1	-	solid
2	-	short dashed
3	-	medium dashed
4	-	long dash short dash
5	-	long dash
6	-	long dash three short dashes
7	-	long dash dot
8	-	long dash three dots
9	-	medium dash dot
10	-	dotted

Each line may have a different type, defined by the number as above. The third attribute is the line width, with larger numbers denoting thicker lines. The fourth number is the marker frequency, denoting the number of points that the user desires to be represented with a marker. (Markers will be discussed in a later section.)

The default for each of these is 1. If the user is satisfied with the default, no LINE attributes need be set. If the user has two lines, then one might type, as an example, the following:

```
LINE = 1;2 / 7;8 / 4;5 / 2
```

This states that the first line use color 1, with line type 7, and a width of 4, and a marker frequency of 2 (every other point). The second line has color 2, line type 8, and width 5, and marker frequency of 2. Note that in the fourth attribute has only one number; that denotes that the number applies to all lines. (Note: This particular option does not work in the current version, version 2007.1 as of this writing. This will be fixed in the next version.) Should a user want to, for example, set only the color and the line width but let the default handle the line type, then the following is typed:

```
LINE = 1;2 // 4;5
```

Note that after the setting for the line colors there are two slashes with nothing in between. That denotes that the user is setting the line color and width, but the section where the line type is blank. Also note that the last section, the marker frequency, is missing. That also tells the program to use the default value of 1. This same method of setting attributes is valid for each option discussed here.

2) MARKER

The marker is the symbol used to distinguish each line. The marker is specified by the following:

```
MARKER=marker color / marker type / size / width
```

As in the LINE option, the MARKER option is also set for each line. Be sure to note that each marker corresponds to the respective line in the LINE option. For example, if there are two lines in the LINE option, there should also be two markers denoted in the MARKER option (unless the user prefers the same marker for both traces), in the same order.

The color attribute is the same as in the LINE option. The size and the width of the marker are determined by a number, with the larger number being the larger size and width. The type is denoted by one of the following:

1	plus sign	12	asterisk
2	octagon	13	hourglass X
3	triangle	14	star
4	box	15	dot
5	small X	16	large X
6	diamond	17	filled octagon

7	up arrow	18	filled triangle
8	X with top bar	19	filled box
9	Z with bar	20	filled diamond
10	Y	21	filled star
11	box with diagonals		

Again, the default value is 1 for all of the options.

3) T_TEXT and L_TEXT

The T_TEXT option allows the user to change aspects of the title of the plot, legend, and axis labels, including titles, fonts, and sizes. This particular option has 3 sections:

```
T_TEXT=TITLE entry | TRACE LABEL entry | AXIS LABEL entry
```

Note that in this case each section is separated from the other sections with a vertical bar (|). Each section has 4 parts: text size, font, width, and hardware flag. Each of these is separated by a slash (/).

The size and the width are represented by numbers as in the other variables. For the font, if the HW flag is used for the hardware font, then the user can select from the following choices:

	REGULAR	ITALIC	BOLD	ITALIC-BOLD
Courier	1	11	21	31
Helvetica	2	12	22	32
Times	3	13	23	33

However, if the flag is SW, then the computer-generated font will be used. Note that for the 3rd section of the T_TEXT, the axis label entry, that if the HW flag is used, then no y-axis will be plotted. FVS is unable to rotate hardware fonts at this time, so in order to print out a y-axis label, it is recommended that SW fonts be used. One can enhance the image of SW fonts with an increase in size or width. However, the above three font options can only be selected with the HW option. The default for this is 1/1/1/SW, for all three sections.

T_TEXT controls only the title, trace legend, and the axis labels. For the tick-mark labels, the user should use L_TEXT, which is the same option except there is only one section rather than three.

4) WITLIN

To see where a given line compares to a threshold (i.e. showing where there is a negative vs. a positive bias), then you can draw a line for that threshold called a witness line. This line can be defined by using the WITLIN option. The option is defined as follows:

```
WITLIN=colr1;...;colrn/typel;...;typen/width1;...;widthn/vall;...;valn
```

The witness line is similar to the LINE option, except that the fourth option is the value of the line the user wants drawn as the witness line. The default for this is 0.

5) XAXIS and YAXIS

The XAXIS option controls the labels on the x-axis. The user defines the starting point, ending point, and the interval of the tick labels. For example:

```
XAXIS=left/right/increment/lbfq:gdfq;tkfq
```

The user can define the starting and ending points of the axis, as well as the increment. The last three elements are the frequencies for labels, grid lines, and tick marks. Here is an example.

```
XAXIS=0/100/5/2;2;2
```

This example shows that the the limits of the x-axis is from 0 to 100, and the plot interval will be every 5. If the user prefers that the code set a default interval, the user can leave off the plot interval, and an appropriate interval will be selected. Every other label will be plotted, as well as every other grid line and every other tick mark. Any elements left blank will force the program to use defaults. This option can also be used for YAXIS.

XAXIS bounds can be set as a time frequency as well. In this case colons (:) are used rather than slashes (/) between the first three elements.

```
XAXIS=t_start:t_stop:increment/lbfq:gdfq;tkfq
```

The starting and ending times are in a YYYYMMDDHH format, and the increment is a number in days, unless the letter Y or M follows the increment (to denote a year or a month increment). For more information, enter “fvs h axis”.

6) TRACEn

This parameter specifies the text of the legend for each line. For example, if this parameter was not set, the text below gives an example of what the legend would look like. These are taken straight from the VSDB file:

```
MODEL=NAM/218
MODEL=RUC/252
MODEL=RTM/255
```

However, in the settings file, the user can set something more informative so it is easier to understand what is being plotted. The n is just a trace number (trace1, trace2, etc...). For

example, the following can be typed

```
/
:
```

```
TRACE1=12-km NAM FORECAST
TRACE2=GSI ANALYSIS
TRACE3=12-km NAM PARALLEL
```

The user should set as many traces as those that the plot will have. These particular labels are more useful to the user to figure out what each line is. The size and font of these trace labels are taken care of in the T_TEXT option; this option only considers the text of the trace labels. If left blank, the default trace titles will be used.

7) XLABEL and YLABEL

These are the text of the labels of the x- and y-axes that are printed rather than the default printout. The font and size of these are taken care of in the T_TEXT option, this only deals with the text. Please note, as described previously, that if HW fonts are used, the y-axis label will not be printed. So only the SW fonts will allow the user to print the text of these axes. If left blank, the default axis labels will be used.

8) TITLE

This is the title of the graph. As with options 6) and 7), the font and size of the title is taken care of in the T_TEXT option. The user can type almost anything. The user-defined title is likely to be more descriptive than the default title. If left blank, the default title will be used. The backslash (\) triggers a new line.

9) DEVICE

This option specifies the graphics device. The output can go either to text, or be plotted in a graph on the screen or in a file. The usage of this command is:

```
DEVICE=device | name | xsize ; ysize | color type
```

The choices of graphics devices are as follows:

TX	Text dump (sizes and color type not applicable)
XW	X Windows terminal
PS	PostScript
NC	NTRANS CGM
GF	GIF file

The name is either the window name or the file name. If the choice is DEVICE=XW, then the name in this case is usually the default, which is GEMPAK. For PS or GF, the user should input an appropriate file name for the respective output type. Note that if PS, NC, or GF are used, no output appears on the screen.

The xsize and ysize represent the dimensions of the output, either in pixels (XW) or in inches (PS). This choice is not applicable to TX or GF. For color type, a flag, applicable only to the PS output device, is used:

C	32 Colors
G	20 Grays
M	Monochrome lines with color fills

In the case of TX, the output will be simply a listing of each point in the plot. It gives the X position and the corresponding value of the Y position. This can be very useful if you want to use the statistical output in another program.

Two Graphs on One Diagram

One can plot two items on the same graph. This following example will demonstrate how to plot both the RMS error and the bias of two models on the same plot. One can apply the following principle to any two items that the user wishes to display on the same graph.

First, run “fvs p”, and then the user gets to Menu 3. Choose option number 2, to explicitly enter trace numbers and corresponding computational codes. Then enter the trace numbers like this: “1 2 1 2” in order to choose the trace numbers. Then at Menu 4, explicitly choose the statistic for each trace. In this example, if the user enters “1102 1102 1101 1101”, then the resulting plot will have four traces, each model having an RMS plot and a bias plot.

Then, at menu 5, the user here must tell the code that a different axis coordinate system is to be plotted on the right side of the graph. The option to do this is RSTRCS. This allows the user to enter the trace numbers, separated by slashes (/), to be plotted using the right-side coordinate system. In this example, the left-side y-axis will have the RMS labels, and the right-side y-axis will have the bias labels. To do this, the choice is RSTRCS=3/4 in this case.

It is not required, however, to have a right-side and a left-side axis to plot both RMS and bias on the same graph. It is perfectly acceptable to use the left-side coordinate to plot both items. However, it would be useful to use the right-side system if the numbers plotted are not of the same order (say, Celsius temperature and sea-level pressure). If the right-side coordinate is not input, this will cause the graph to be stretched out. When this happens, it can be difficult to detect trends in the lines.

When the right-side coordinate is used, the right-side labels can also be input. The analogs to WITLIN, YLABEL, and YAXIS are RWTLIN, YRLBEL, and RYAXIS. These three options

are used in the same manner as the left-side options described in the previous section.

More Than One Plot on a Page

Another useful tool is the ability to plot several plots onto one page, side-by-side or one above the other. This can be done by using the PANEL option from Menu 5. The user can plot up to 9 plots on a page.

Below is a list of options the user can define:

NUMBER	ABBREVIATION	DESCRIPTION
0	ALL	Entire device
1	UL	Upper left quadrant
2	UR	Upper right quadrant
3	LL	Lower left quadrant
4	LR	Lower right quadrant
5	L	Left half
6	R	Right half
7	T	Top half
8	B	Bottom half

So for example, to plot a graph in the upper left quadrant of a page, then the option to use is PANEL=1, or PANEL=UL. Then it is possible to plot more graphs on a page in the other three quadrants. Be sure to specify CLEAR=NO between each plot, otherwise the previous plots will disappear.

If the user wishes to plot 9 graphs to a page, then use the following options, depending on the desired location of the graph:

M1	M2	M3
M4	M5	M6
M7	M8	M9

Yet another option would be to divide the page horizontally or vertically into thirds or fourths. Then the user would use PANEL=Tij, where T is either H for horizontal and V is vertical, i is 3 for thirds or 4 for fourths, and j is the panel number counting from the left or from the top.

Finally, the user may simply specify the desired location of the plot using the lower left and the upper right corners of the plot, in fractions of the page. The origin, (0,0) of the page is the lower left corner. For example, to plot the lower left quadrant:

```
PANEL=0;0;.5;.5
```

This indicates that the lower-left corner of the plot is at the origin, (0,0), and the upper-right corner is halfway across the page and halfway up the page (0.5 of a page).

ROC Diagram

Most performance measures are simple to plot using the methods shown above. One statistic, known as the Relative Operating Characteristic, or ROC, diagram, provides a method to evaluate the probability of an event to occur. It is calculated using FHO statistics and requires overlaying one plot on top of another. Unlike a plot such as the Equitable Threat Score, which is provided to show statistics for all thresholds on one plot, each ROC diagram shows a statistic for one individual threshold. So if there are, for example, 8 thresholds, then the user needs 8 ROC diagrams to describe each individual threshold.

Let's start with an example of a trace.ctl file that is used to plot a ROC diagram. This defines a time series of FHO values for a given threshold.

```
1 5 1 2 1 1 1 0/0/0/0
15 0 0 200612150000 200701150000
1 2 0 2 1 1 1 0/0/0/0
18 0 0 NAM/218 0
1 3 0 2 1 1 1 0/0/0/0
9 24 0 0 0
1 9 0 2 1 1 1 0/0/0/0
18 0 0 G236 0
1 10 0 2 1 1 1 0/0/0/0
18 0 0 FHO 0
1 11 0 2 1 1 1 0/0/0/0
9 5 0 0 0
1 12 0 2 1 1 1 0/0/0/0
18 0 0 DPT 0
```

This is a small trace.ctl file for a one-month time period, calculating FHO statistics for the 5 degree threshold of dew point. Do an “fvs s” search as usual, and then run “fvs p”.

Hit “1” for Menu 1. Then for Menu 2, choose “2” to plot a scatterplot. On Menu 3, hit “2” to explicitly enter trace numbers. For those trace numbers, simply type “1 1”. Then on Menu 4 (the FHO options menu), the numbers to type are “1021 1007”. This will produce the scatterplot of False Alarm Rate vs. Probability of Detection (also known as the Hit Rate). Then, at Menu 5, choose the following options to customize the plot:

```
yax=-.02/1.02/.1
xax=-.02/1.02/.1
yla=HIT RATE/2
xla=FALSE ALARM RATE/2
lablev=2
line=0
mark=23/1/1/2
```

Note here that shortened versions of some of the options discussed earlier are used (such as “yax” rather than “yaxis”, “yla” rather than “ylabel”, “mark” rather than “marker”, etc.).

After the above items are entered, type “r” to run the plot. The scatterplot comes out, with the hit rate on the y-axis and the false alarm rate on the x-axis. The various values are plotted on the chart. However, this does not complete the plot.

The next step is to plot the ROC curve itself. This is done by running “fvs p” as usual, but then at Menu 2, the option “5”, for displaying ROC curves, should be selected. Then once the user gets to Menu 4, the code “1032” for the ROC curve should then be selected.

Then the very same items as above should be input at Menu 5 to complete the plot, with the following exceptions:

```
clear=no
mark=0
```

Add a title (using the TITLE option), and the new plot is overlaid onto the scatterplot. The result is the ROC diagram that can be used to assess how accurate the forecast is for that particular threshold. To get a complete suite of plots, this entire process should be repeated for each individual threshold.

FVS in Scripts

To many users it is arduous and time-consuming to manually use “fvs p” to plot a graph, especially if there are many graphs to plot on a regular basis. The options to plot can be selected on the fly rather than pre-selected. This allows users to put their fvs graphing commands in a script.

For example, suppose that a time range is set with a date put in a variable DATE1 for the starting date, and DATE2 for the ending date, 30 days after DATE1. Suppose that a time series is desired, and two traces are requested. For example, suppose the user wants to plot a NAM time series versus the GFS, RUC, NAMX, or NAMY forecasts, for each 12 hours of the forecast up to 48 hours. Also, assume that the user wants to plot both RMS and bias plots. Your script would look something like this.

```
for hr in 00 12 24 36 48
do

for model in GFS/212 RUC/252 NAMX/218 NAMY/218
do

cat <<EOF> trace.ct1
1 5 1 2 1 1 1 0/0/1/0
15 0 0 ${DATE1} ${DATE2}
2 5 1 2 1 1 1 0/0/1/0
15 0 0 ${DATE1} ${DATE2}
1 1 0 2 1 1 1 0/0/1/0
18 0 0 V01 0
```

```

2 1 0 2 1 1 1 0/0/1/0
18 0 0 V01 0
1 2 0 2 1 1 1 0/0/1/0
18 0 0 NAM/218 0
2 2 0 2 1 1 1 0/0/1/0
18 0 0 $model 0
1 3 0 2 1 1 1 0/0/1/0
9 $hr 0 0 0
2 3 0 2 1 1 1 0/0/1/0
9 $hr 0 0 0
1 8 0 2 1 1 1 0/0/1/0
18 0 0 ONLYSF 0
2 8 0 2 1 1 1 0/0/1/0
18 0 0 ONLYSF 0
1 9 0 2 1 1 1 0/0/1/0
18 0 0 G236 0
2 9 0 2 1 1 1 0/0/1/0
18 0 0 G236 0
1 10 0 2 1 1 1 0/0/1/0
18 0 0 SL1L2 0
2 10 0 2 1 1 1 0/0/1/0
18 0 0 SL1L2 0
1 12 0 2 1 1 1 0/0/1/0
18 0 0 T 0
2 12 0 2 1 1 1 0/0/1/0
18 0 0 T 0
1 14 0 2 1 1 1 0/0/1/0
18 0 0 SFC 0
2 14 0 2 1 1 1 0/0/1/0
18 0 0 SFC 0
EOF

```

fvs s

```

cat << EOF > settings1.ct1
DEVICE = xw|VSDB
CLEAR = yes
L_TEXT=1.5/21/1.25/HW
TITLE=${hr}-hr forecast of Temperature
line=2;4/1;2/6;6
T_TEXT=1.5/21/1.25/HW|1.5/21/1.25/HW|1.5/21/1.25/SW
ylabel=2-m Temp RMS error
xlabel=Forecast Hour
tracel=NAM
trace2=${model}
EOF

```

```

cat << EOF > settings2.ct1
DEVICE = xw|VSDB
CLEAR = yes
L_TEXT=1.5/21/1.25/HW
TITLE=${hr}-hr forecast of Temperature
line=2;4/1;2/6;6
T_TEXT=1.5/21/1.25/HW|1.5/21/1.25/HW|1.5/21/1.25/SW
ylabel=2-m Temp Bias
xlabel=Forecast Hour
tracel=NAM
trace2=${model}
EOF

```

```

fvs p <<ENDFVS
1
1
1
1102
0
0

```

```

a settings1.ct1
r
cp plot1.ps
0
0
1
1
1
1101
0
0
a settings2.ct1
r
cp plot2.ps
0
0
ENDFVS

done

done

```

The script for an upper-air plot may look like this:

```

for hr in 00 12 24 36 48
do

for model in GFS/212 RUC/252 NAMX/218 NAM/218
do

cat <<EOF> trace.ct1
1 14 1 2 8 1 4 0/0/1/0
18 0 0 P1000 0
18 0 0 P850 0
18 0 0 P700 0
18 0 0 P500 0
18 0 0 P400 0
18 0 0 P300 0
18 0 0 P250 0
18 0 0 P200 0
2 14 1 2 8 1 4 0/0/1/0
18 0 0 P1000 0
18 0 0 P850 0
18 0 0 P700 0
18 0 0 P500 0
18 0 0 P400 0
18 0 0 P300 0
18 0 0 P250 0
18 0 0 P200 0
1 1 0 2 1 1 4 0/0/1/0
18 0 0 V01 0
2 1 0 2 1 1 4 0/0/1/0
18 0 0 V01 0
1 2 0 2 1 1 4 0/0/1/0
18 0 0 NAM/218 0
2 2 0 2 1 1 4 0/0/1/0
18 0 0 $model 0
1 3 0 2 1 1 4 0/0/1/0
9 $hr 0 0 0
2 3 0 2 1 1 4 0/0/1/0
9 $hr 0 0 0
1 5 0 2 1 1 4 0/0/1/0
15 0 0 ${DATE1} ${DATE2}
2 5 0 2 1 1 4 0/0/1/0
15 0 0 ${DATE1} ${DATE2}
1 8 0 2 1 1 4 0/0/1/0

```

```

18 0 0 ADPUPA 0
2 8 0 2 1 1 4 0/0/1/0
18 0 0 ADPUPA 0
1 9 0 2 1 1 4 0/0/1/0
18 0 0 G236 0
2 9 0 2 1 1 4 0/0/1/0
18 0 0 G236 0
1 10 0 2 1 1 4 0/0/1/0
18 0 0 SL1L2 0
2 10 0 2 1 1 4 0/0/1/0
18 0 0 SL1L2 0
1 12 0 2 1 1 4 0/0/1/0
18 0 0 T 0
2 12 0 2 1 1 4 0/0/1/0
18 0 0 T 0
EOF

```

```
fvs s
```

```

cat << EOF > settings1.ct1
DEVICE = xw|VSDB
CLEAR = yes
revers=y
L_TEXT=1.5/21/1.25/HW
TITLE=${hr}-hr forecast of Temperature
line=2;4/1;2/6;6
T_TEXT=1.5/21/1.25/HW|1.5/21/1.25/HW|1.5/21/1.25/SW
ylabel=2-m Temp RMS error
xlabel=Forecast Hour
tracel=NAM
trace2=${model}
EOF

```

```

cat << EOF > settings2.ct1
DEVICE = xw|VSDB
CLEAR = yes
revers=y
L_TEXT=1.5/21/1.25/HW
TITLE=${hr}-hr forecast of Temperature
line=2;4/1;2/6;6
T_TEXT=1.5/21/1.25/HW|1.5/21/1.25/HW|1.5/21/1.25/SW
ylabel=2-m Temp Bias
xlabel=Forecast Hour
tracel=NAM
trace2=${model}
EOF

```

```

fvs p <<ENDFVS
1
1
1
1102
0
0
a settings1.ct1
r
cp plot1.ps
0
0
1
1
1
1101
0
0
a settings2.ct1
r

```



```

cp plot2.ps
0
0
ENDFVS

done

done

```

These are not very much different, except the `revers=y` option for the upper-air profile, which is necessary. Most any type of plot or plotting option described above can be input into the above. Most important seems to be the `fvs p <<ENDFVS` part of the script. This runs the `fvs p` plotter and also inputs all the options that the user needs. All the inputs need to be known ahead of time in order to put in the script, so it is useful to plot each plot manually ahead of time so all the options are known. A missing or mis-written option will result in an error in the program and no plots being plotted.

The user can also do a script for the diurnal/model cycle of two forecasts versus one set of observations, for several models:

```

cyc=00
# cyc=12

if [ $cyc -eq 00 ]; then
set -A hrs 00 12 00 12 00 12 00 12 00 12
elif [ $cyc -eq 12 ]; then
set -A hrs 12 00 12 00 12 00 12 00
fi

for model in GFS/212 RUC/252 NAMX/218 NAMU/218
do

cat <<EOF> trace.ct1
1 3 1 2 8 1 4 0/0/1/0
9 00 0 0 0
9 12 0 0 0
9 24 0 0 0
9 36 0 0 0
9 48 0 0 0
9 60 0 0 0
9 72 0 0 0
9 84 0 0 0
2 3 1 2 7 1 4 0/0/1/0
9 00 0 0 0
9 12 0 0 0
9 24 0 0 0
9 36 0 0 0
9 48 0 0 0
9 60 0 0 0
9 72 0 0 0
9 84 0 0 0
1 15 1 2 8 1 4 0/0/1/0
18 0 0 ${hrs[0]}00 0
18 0 0 ${hrs[1]}00 0
18 0 0 ${hrs[2]}00 0
18 0 0 ${hrs[3]}00 0
18 0 0 ${hrs[4]}00 0
18 0 0 ${hrs[5]}00 0
18 0 0 ${hrs[6]}00 0

```

```

18 0 0 ${hrs[7]}00 0
18 0 0 ${hrs[8]}00 0
2 15 1 2 8 1 4 0/0/1/0
18 0 0 ${hrs[0]}00 0
18 0 0 ${hrs[1]}00 0
18 0 0 ${hrs[2]}00 0
18 0 0 ${hrs[3]}00 0
18 0 0 ${hrs[4]}00 0
18 0 0 ${hrs[5]}00 0
18 0 0 ${hrs[6]}00 0
18 0 0 ${hrs[7]}00 0
18 0 0 ${hrs[8]}00 0
1 2 0 2 1 1 4 0/0/1/0
18 0 0 NAM/218 0
2 2 0 2 1 1 4 0/0/1/0
18 0 0 $model 0
1 5 0 2 1 1 4 0/0/1/0
17 0 0 ${DATE1} ${DATE2}
2 5 0 2 1 1 4 0/0/1/0
17 0 0 ${DATE1} ${DATE2}
1 8 0 2 1 1 4 0/0/1/0
18 0 0 ONLYSF 0
2 8 0 2 1 1 4 0/0/1/0
18 0 0 ONLYSF 0
1 9 0 2 6 1 4 0/0/1/0
18 0 0 G236 0
2 9 0 2 6 1 4 0/0/1/0
18 0 0 G236 0
1 10 0 2 1 1 4 0/0/1/0
18 0 0 SL1L2 0
2 10 0 2 1 1 4 0/0/1/0
18 0 0 SL1L2 0
1 12 0 2 1 1 4 0/0/1/0
18 0 0 T 0
2 12 0 2 1 1 4 0/0/1/0
18 0 0 T 0
1 14 0 2 1 1 4 0/0/1/0
18 0 0 SFC 0
2 14 0 2 1 1 4 0/0/1/0
18 0 0 SFC 0
EOF

```

```
fvs s
```

```

cat << EOF > settings1.ct1
DEVICE = xw|VSDB
CLEAR = yes
L_TEXT=1.5/21/1.25/HW
TITLE=2-m Temperature Cycle: Obs vs. Forecast
line=2;4;3/1;2;3/6;6;6
T_TEXT=1.5/21/1.25/HW|1.5/21/1.25/HW|1.5/21/1.25/SW
ylabel=2-m Temp Bias
xlabel=
tracel=NAM
trace2=${model}
trace3=Observations
xudlbl=00/12/24/36/48/60/72/84
EOF

```

```

fvs p <<ENDFVS
1
1
2
1 2 1
2 2 3
0
0

```

```
a settings1.ctl
r
cp plot1.ps
0
0
ENDFVS

done
```

Miscellaneous information

One other thing that has not been noted until this point. A trace.ctl file does not have to contain only one control plot. It is possible to have more than one plot defined in a single trace.ctl file. It should be noted that the user must be very careful to use the correct plot number for each plot in the header line. Also when doing this, the user also needs to note that the number of traces in each plot does not reset when starting a new plot in the same file. For example, if plot 1 has three traces and plot 2 has two traces, plot 1 will have traces 1 through 3 and plot 2 will have trace numbers 4 and 5. However, it is not possible to plot, in this example, traces 1 and 4 on the same plot. It is recommended to use the “fvs u” option to create such a control file, if only for the first time it is done.

Also note that when a user runs the fvs p program and completes a plot, fvs gives the user a chance to enter in a new control file other than having the name “trace.ctl” file to do a new plot. If this is done, it is advisable for the user to do an fvs s on each control file and save the resulting trace.dat file, the actual data that the fvs plotting codes uses to calculate the statistics. This would be useful should there be an interest in plotting the same data at a later time without needing to run the search code.

Summary

FVS is a useful tool that plots out a wide variety of statistics from different sources. It is a flexible tool but also one that has many options and can be confusing. It is highly recommended that the user run “fvs u” first. However, it is possible to create the information needed for producing a plot by following the instructions above. Time series, upper-air profiles, scatterplots, model cycles, FHO statistics, tropical cyclone statistics, and ensemble statistics nearly comprise the entirety of possible verification plots for most model runs.

FVS is always in development, so new features are continually being added. This document will also undergo changes as FVS capabilities undergo changes, so be sure to look for the latest updated document (see the date at the very top of the page).