Version 1.3 May 2020 Gwyn Griffiths gwyn at autonomousanalytics.com

See separate document for use of Grafana with Influx - in case it is of use to someone

Creating Grafana dashboards with live connection to Timescale wsprdaemon databases

New in this version: Typo corrections and examples of SQL-only queries.

Contents

0. Installation and initial connection

Grafana is available for Windows, Mac, and Linux (including ARM) operating systems. Download and installation details are at:

https://grafana.com/grafana/download

There is no need to make any changes to the configuration options to run Grafana or to configure and install the dashboards covered in this note.

Connect to your own Grafana installation via a web browser to:

http://localhost:3000

Login with the default userid and password - both 'admin'. Then change your password.

If, however, you would like to try before you install your own version you are welcome to use the Grafana installation at http://wsprdaemon.org:3000
where you can log in as user open and password open
with access to two dashboards:

Open Access Noise Graph and 20m at Northern Utah
to illustrate Grafana's capability with wsprdaemon data.

The Open Access Noise Graph dashboard lets you choose the site_name, receiver_name, band_m and noise_type from the pull-down menus. You can also change the time interval.

The 20m at Northern Utah dashboard shows you a range of time series graphs appropriate to wspr spots data. What you can't do as user open is save your changes or create new dashboards.

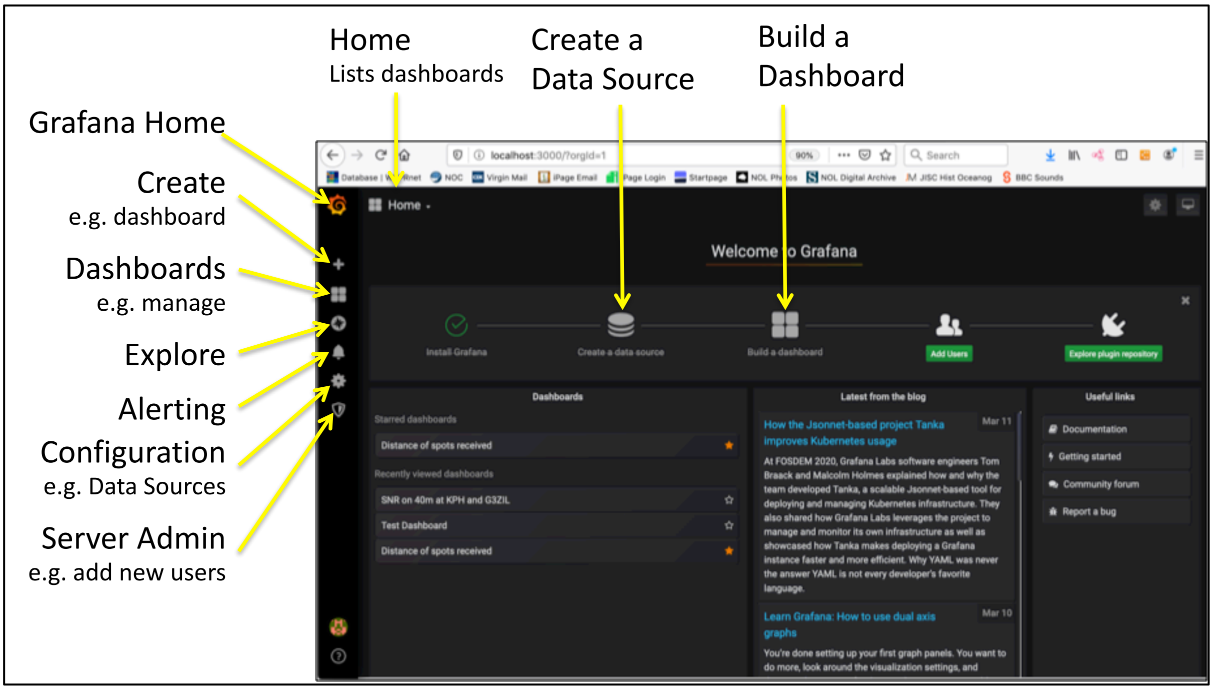
Version 1.3 May 2020 Gwyn Griffiths gwyn at autonomousanalytics.com

See separate document for use of Grafana with Influx - in case it is of use to someone

wsprdaemon users are welcome to use the Grafana installation at http://wsprdaemon.org:3000 where you can log in as user wdread and request a password (if not known to you) from gwyn at autonomousanalytics.com where you will have 'edit' rights on copies of the Open Access Noise Graph and 20m at Northern Utah dashboards for you to see how they are constructed. You have permission to save your changes, or better, use save as; you can also create and save new dashboards. With this experience, you may then want your own installation.

The following notes are for those with their own installation, but sections 2 onwards are applicable to those with wdread access to at http://wsprdaemon.org:3000.

On login you should see the Home screen, below, showing you have installed Grafana, although in this example we already have a data source connected and some example Dashboards.



How the Jsonnet-based project Tanka improves Kubernetes usage Mar 11

At FOSDEM 2020, Grafana Labs software engineers Tom Braack and Malcolm Holmes explained how and why the team developed Tanka, a scalable Jsonnet-based tool for deploying and managing Kubernetes infrastructure. They also shared how Grafana Labs leverages the project to manage and monitor its own infrastructure as well as showcased how Tanka makes deploying a Grafana instance faster and more efficient. Why YAML was never the answer YAML is not every developer's favorite language.

Learn Grafana: How to use dual axis graphs Mar 10

You're done setting up your first graph panels. You want to do more, look around the visualization settings, and

1. Create a data source

From the screen above (click Grafana Home if you have explored some of the other screens), click on Create a data source. Scroll to PostgreSQL and select, bringing up the Settings page.

Give the data source a name, e.g. wsprdaemon_spots and click on the slider to set to your default.

Enter the Host as wsprdaemon.org:5432

Enter Database as tutorial

Enter User as wdread and password as JTWSPR2008

For SSL Mode select disable

Leave Connection Limits as they are

Scroll down to the section PostgreSQL details (on my browser the scroll bar on the right is both very thin and a shade of grey just off-black making it hard to see).

See separate document for use of Grafana with Influx - in case it is of use to someone

For Version select 10 in the drop-down list (if 11 appears, select it instead).

Slide the TimescaleDB switch to the right. Leave Min time interval at 1m.

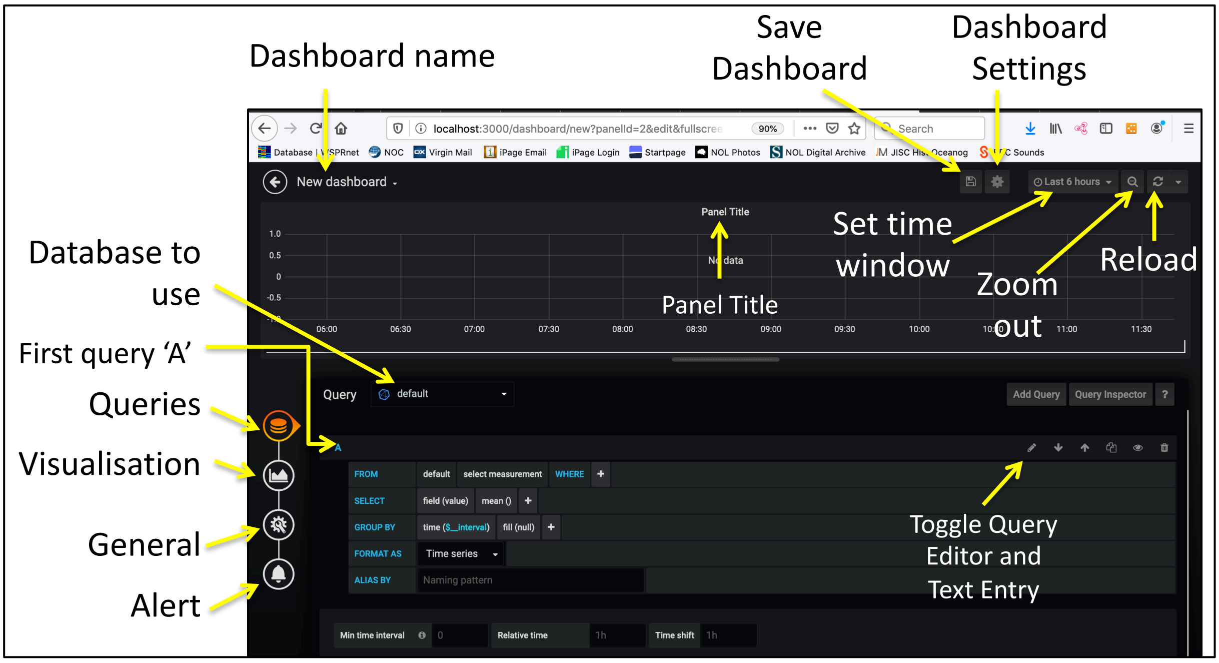
Scroll to bottom of the screen and click the green Save & Test button. All being well you should see a green banner with a tick and Database Connection OK message.

2. Create a dashboard

Hover over the Create icon at top left (big plus), click on Dashboard.

2.1 Create the first Panel.

Click on Add Query, which will bring up the screen below.



FROM default select measurement WHERE +

SELECT field (value) mean () +

GROUP BY time ($__interval) fill (null) +

FORMAT AS Time series

ALIAS BY Naming pattern

Min time interval 0 Relative time 1h Time shift 1h

The upper part of the screen shows a default time series graph - a Panel in Grafana jargon.

The lower part is a form-filling query builder to collect data from the Timescale DB.

The query builder is ready for Query 'A', others can be added later. As a first example, let us plot the SNR for N5CNB at KPH on 20m:

First, next to Query, pull down the database name e.g. wsprdaemon_spots.

On the FROM line click on default and from the pull-down options select spots.

Leave Time column as "time". After Metric column select tx_call from the pull down menu

On the SELECT line click, it may already show Column: "SNR", If it does not, click on whatever text is after Column: and select SNR from the pull-down. on value (within the brackets) and select SNR.

Click on the + to the right of WHERE, in the empty box, click and you may see a pull down list, or you may not, type E and it may come up with the option Expression, if not type it. At this point a red error banner may come up, that's ok, we've not set values yet. Click on the first value and select rx_id (may not always 'take', if so try selecting again, if still nothing, type in 'KPH', the single quotes are needed). Click on the plus again, and in the same manner, click and select in turn Expression: band = '20'. Click on the plus again, and in the same manner, click and select in turn Expression: tx_call = 'N5CNB'.

At this point, data should appear. If there is no data, try zooming out in time using the zoom out icon (magnifying glass with -).

Save your work so far by clicking on the disk icon to Save Dashboard. Give the Dashboard a suitable name, e.g. SNR at KPH, noting that we'll be adding further Panels to this Dashboard and so the Dashboard name should be appropriate.

2.2 Add a second (and subsequent) time series to this Panel.

After a save the Query Builder will have been left, if so click on the Panel Title then on Edit, which will bring back the query builder.

Click on the Add Query button at centre right, which will bring up a query builder for query 'B'. Of course, this second query should be along the same lines at the first, e.g. still SNR at KPH on 20m but, say, for N6WKZ.

2.3 Customise the Panel.

Click on the Visualisation (graph-like) icon on the left. Under the Draw Modes column you might want to use points rather than lines, if so, slide the lines switch to the left and the points switch to the right. Under the Mode Options column you can set line fill and thickness or the radius of points.

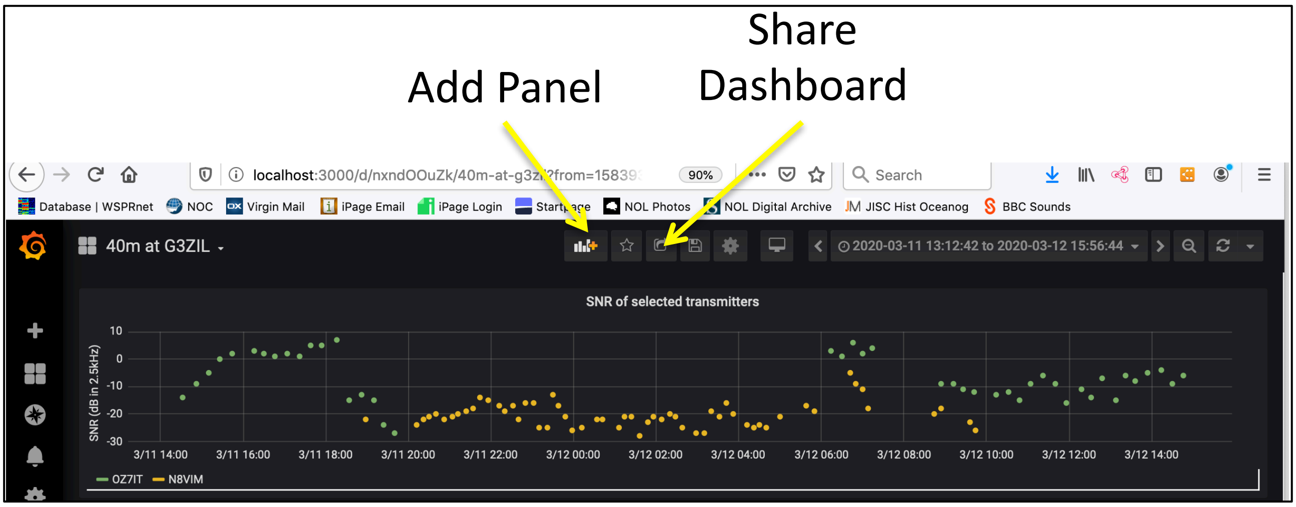
Scroll down to Axes, under the Left Y column you might want to set fixed Y-min and Y-max, and Decimals (for number of decimal places). You should also enter a Label for the Y-axis, e.g. SNR (dB in 2.5kHz). There's no need to alter Legend settings.

Click on the disk icon at the top to save the revised Panel.

2.4 Add a second Panel

Click on the Dashboard title at top left. In the list of Dashboards select the one where you want to add a panel. This brings up a few other icons as in the screenshot below.

Click on Add Panel (the bargraph icon), then Add Query, and the process is essentially as described for the first Panel. However, we can demonstrate here some additional graphical forms within Grafana, in this case a bargraph grouped by time.



The theme is still 20 m at KPH for the WHERE clause, but for this panel we will show the number of spots in each 10 minute interval.

To the right of Query select wsprdaemon_spots as the data source.

On the FROM line select spots.

On the SELECT line click on the + and you should see as an option Aggregate functions, and to its right, count. Data should be seen in the new Panel at this stage. (Column:"SNR") should already be present, and then Alias: "SNR" should automatically appear after count is selected, click on the "SNR" and type in the proper descriptor, Spots in 10 minutes, which should then appear as the text alongside the key at the bottom left of the graph).

On the GROUP BY line click within the brackets after time, and select 10m (i.e. 10 minutes).

Click on the Visualisation icon at left, and you may want to change to Bars rather than Lines in the Draw Modes column.

Scroll down to Axes and and for the Left Y axis enter a Label such as Spots in 10 minutes.

Click on the General icon at left, and enter a Panel title such as Spots in 10 minutes.

Click on the disk icon at the top to save.

Click on the Dashboard name top left, then the selected Dashboard to resize, and adjust time range to suit.

2.5 Add a third Panel.

Grafana has a simple form of 'heatmap' graph format. We can use it, for example, to show the variation with time of the spread of distance of received spots. Following the Add Panel procedure above for the FROM line, select spots. On the SELECT line click the Column variable and select 'km'. On the WHERE line click the Macro: option and select remove. Click the +, select Expression, then select 'rx_id' and then type in 'KPH', click the +, select Expression, then select 'band' and then select or type in '20'. If there is an entry on the GROUP BY line, select the function and select remove.

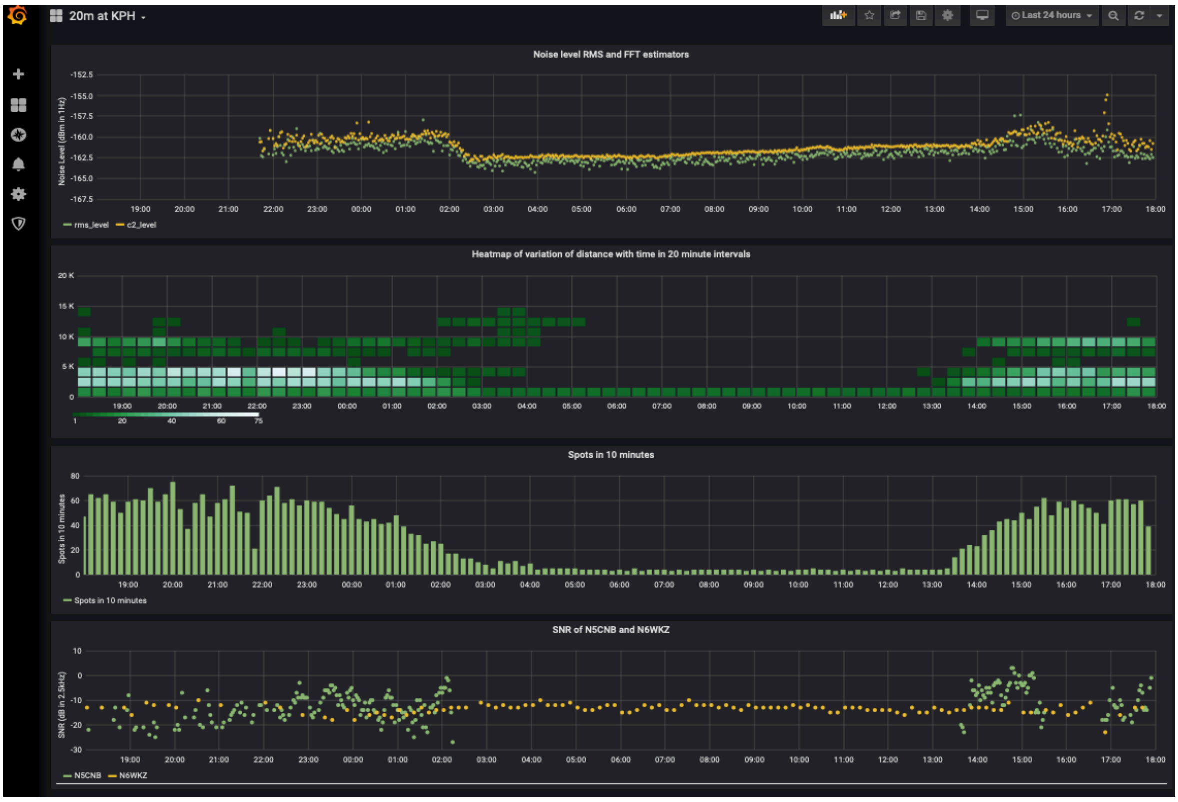
Click the Visualisation icon, next to Visualisation in the Graph pull down, select Heatmap. You may want to set a suitable Y-min and Y-max under the Visualisation Y axis column. Under the Buckets column you may want to experiment to set the number of buckets for the x and y axes by number or by units (e.g. in this case km for y, and perhaps 20m for x). Under the Display options you may want to experiment with the Scheme - the color scale to use. Under the Legend column select the Show Legend option for a scale bar. Click General on left to give the Panel a title. Click the disk icon to save. We now have a complete example Dashboard as below.

2.6 Add a fourth Panel

From wsprdaemon version 2.8a rms and c2_FFT noise estimates are sent to the Timescale database on the wsprnet.org server. Noise graphs can therefore be presented alongside spots data. Following the steps as above, but on the FROM line select noise as the data source. On the SELECT line, Column: rms_level may already be present, click +, select Column and then select c2_level. On the WHERE line click Macro: and select remove, click +, then Expression, band and 20, click +, then Expression, receiver, KPH_HF_78.

Click the Visualisation icon, under Draw Modes, Points are best, and because of the 2-minute interval data, under Mode Options select Point Radius 1. Under Axes you may want to set limits on the Left Y axis, and add a Label e.g. Noise Level (dBm in 1Hz).

The Dashboard we have created should look like the screenshot below. Having gone through this tutorial for three different graph forms you should now have enough working knowledge to create your own Panels and Dashboards.



2.7 Change Dashboard Settings

Click the Dashboard Settings (cog) icon at the top of the screen.

Under Time Options you can select the Timezone to be determined by the local browser time or UTC. I suggest using UTC.

For Auto-refresh set a value such as 5m.

For Now delay now set a value of 1m to ignore last minute, which may have partial data.

Click Save. You could also click Save As to use this Dashboard as a template for others, e.g. for different bands.

Click on Dashboard name top left to get back to list of Dashboards.

3. Creating a Panel with pull-down selections

The Dashboard and panels described above have had the variable field selections to plot 'hard-coded' into the Query Builder and hence the plots themselves. Using the Template and Variables features in Grafana we can construct panels with pull-down selections to enable more general-purpose plots, along the lines of the original Grafana noise plots produced by Tommy Nourse KI6NKO. Our example will be a noise measurements panel.

3.1 Pull-down selections within the WHERE clause

On opening Grafana click on Build a Dashboard, then Add Query. Click the Dashboard settings cog icon at top right, give the Dashboard a name; in Time options set Timezone to UTC, Auto refresh to 5m and Now delay to 1m.

On the menu list at top left click on Variables, click on Add Variables. For the noise example we need three variables: Site, Receiver and Band, we'll call then site_name, receiver_name and band_m. Under the General heading, enter the Name - site_name. Under Query options, Data source, select wsprdaemon_noise, for Refresh, select On Time Range Change.

The next step is to code the SQL query that will generate the list of site names for the pull-down list. In the Query box enter:

select distinct site from noise

From the noise table this selects the distinct (unique) site names.

Set Sort as Alphabetical (asc), move the Multi-value slider to the right, to select more than one site to plot. You'll see a preview list of unique site names. Click Add then Save. Click Dashboard settings, top right, then Variables top left, click Duplicate at right against the site_name entry. Click on the copy, rename to receiver_name.

Edit the Query box to read:

select distinct receiver from noise where site in ($site_name)

As well as selecting distinct receiver names, we are specifying only those receiver names applicable to the sites we have already selected[1]. Again, move the Multi-value slider to the right. You may well see error messages in a red box, ignore them. Click Update.

Click Duplicate on the receiver_name line, click on the copy line, as the Variables>Edit page comes up, change the Name to band_m and edit the query box to read:

select distinct band from noise where site in ($site_name)

Set Sort order to Numerical (asc). Click Save. Click the Go Back left arrow, top left. Click the Panel Title and select Edit, which will bring up the Query builder. Select wsprdaemon_noise as data source. For Metric pull-down select receiver. On the Select line click the + and select Column, click to select c2_level if a second rms_level has appeared. On the Where line click the + then select Expression, pull down the first value to be site, and for the second value, select $site_name (i.e. the variable, and not a fixed name). Click + again for Expression, receiver, $receiver_name, + again, for Expression, band, $band_m. Error messages may appear, ignore for now.

In the Format line, click on Edit SQL, the SQL may well look like the following:

```
SELECT
  "time" AS "time",
  receiver AS metric,
  rms_level,
  c2_level
FROM noise
WHERE
  site = '$site_name' AND
  receiver = $receiver_name AND
  band = $band_m
ORDER BY 1,2
```

There is a problem here - the Query Builder hasn't formed the query correctly for postgreSQL, the use of "=" in the WHERE clause is appropriate for a specific site, receiver or band, but not for a list. For a list the correct syntax is WHERE site IN ($site_name); so IN instead of =, and parentheses instead of single quotes. So edit the SQL to read:

```
SELECT
  "time" AS "time",
  receiver AS metric,
  rms_level,
  c2_level
FROM noise
WHERE
```

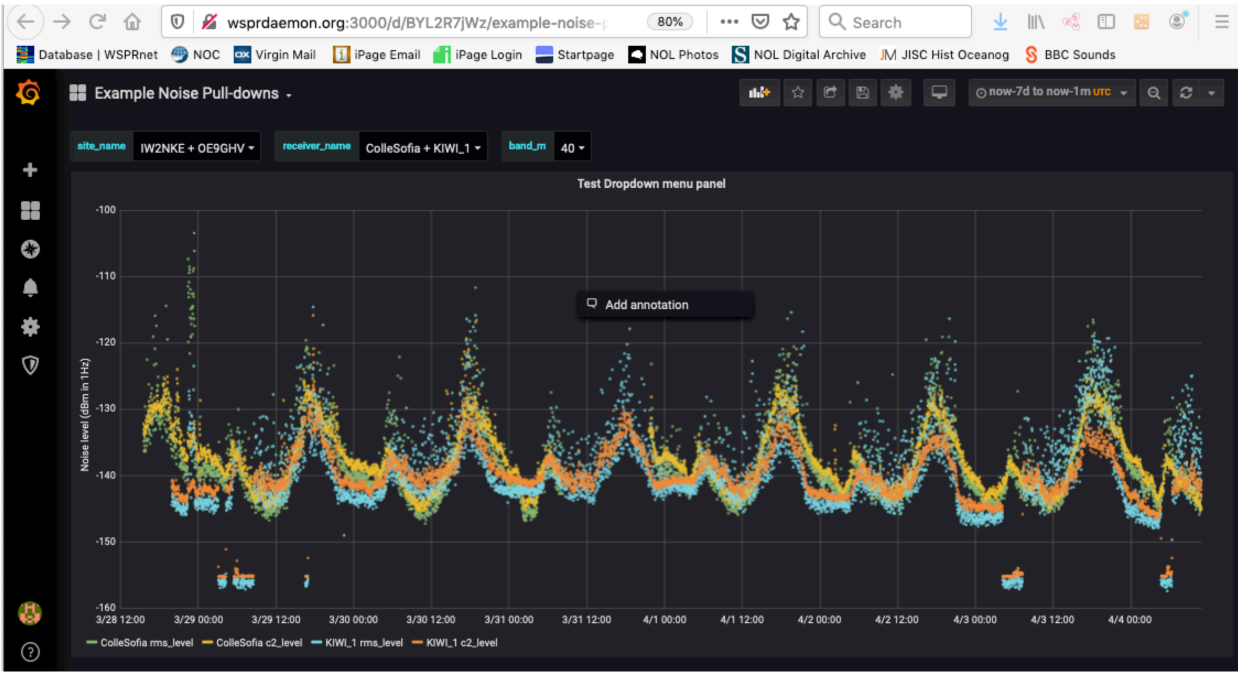
[1] This helps minimise confusion as there are many receiver names in the list with varying styles.

```
  site IN ($site_name) AND
  receiver IN ($receiver_name) AND
  band IN ($band_m)
ORDER BY 1,2
```

We should now see the three pull downs properly populated with options and some data graphed. Click on the Visualisation icon, middle left, select points rather than lines, and set Point Radius to 1. Scroll down to Axes and add Y axis label: Noise level (dBm in 1Hz).

Click the General icon, middle left,, give the Dashboard a title, click the disk save icon at top, saving current time range and current variables.

If your expected pull down options do not appear, click on the graph panel, or the refresh icon at far top right and wait a few seconds. Here is the finished panel displaying data from OE9GHV and IW2NKE on 40m.



wsprdaemon.org:3000/d/BYL2R7jWz/example-noise-

Example Noise Pull-downs

site_name: IW2NKE + OE9GHV

receiver_name: ColleSofia + KIWI_1

band_m: 40

now-7d to now-1m UTC

3.2 Pull-down selections within the SELECT clause

We can also create pull-down selection lists for a choice of column names, for example to select c2_level or rms_level or both from the noise Table. In this case we need the variable for the selection to be accessible in the SELECT clause within the Query Builder. This is rather more complicated and not at all intuitive.

Continuing with the example noise Table as in 3.1, follow the process to create a newVariable as in that section; let us call the variable noise_type. In Query Options, set Data source as before, Refresh On Time Range Change, and in the Query line use the following to list the column names:

SELECT json_object_keys(to_json((SELECT t FROM public.noise t LIMIT 1)));

Alone, this will list us all the column names (see the Preview of values at bottom), but we just want c2_level and rms_level, so in the Regex box enter

/level/

to only list the output from the SELECT that includes 'level'. Select Sort as alphabetical (ascending). In Selection Options slide Multi-value on. Save the Variables and then go back to the Dashboard and pull down edit from its title so we can return to the Query Builder.

The SELECT clause in the Query Builder needs to be changed to make use of the variable noise_type. In needs to change to the following, **note** the curly braces in ${noise_type:csv}:

```
SELECT
 "time" AS "time",
  receiver AS metric,
  ${noise_type:csv} as value
FROM noise
WHERE
  site IN ($site_name) AND
  receiver IN ($receiver_name) AND
  band IN ($band_m)
ORDER BY 1,2
```

It took me quite some time to find the correct syntax for ${noise_type:csv}. The curly braces and :csv signify "disable quoting" and produces a comma separated list.

3.3 Queries using SQL view only, not the Query Builder

There are queries we may want to use that are not supported by the Query Builder options. Grafana allows direct entry of all of the SQL statements needed to form a query. In this example we have a mix of three direct entry and one Query Builder queries to form a panel to show spot distance statistics in 10 minute intervals. The example is in the wsprdaemon community folder as Community HF Distance Stats. Note it can take a long time to run sometimes ...

The one Query builder query (B) is for the average - from the pull-down menu as an aggregate function. The other statistics we want are the median, lower quartile and upper quartile, where the following SQL is entered in queries A, C and D:

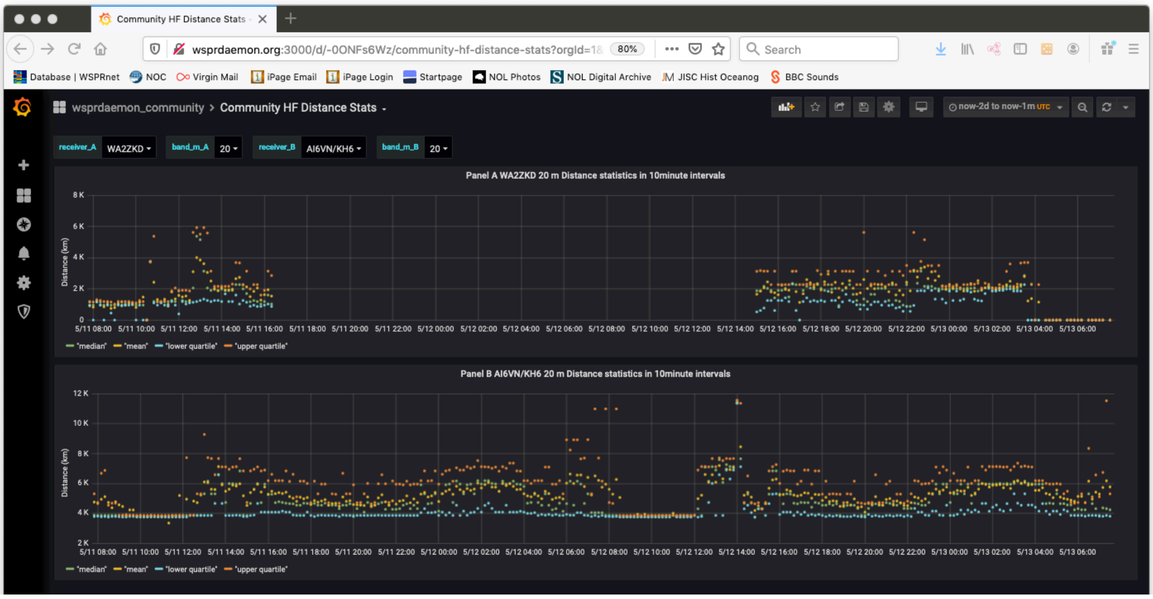
```
SELECT
  $__timeGroupAlias("time",10m),
    percentile_disc(0.5) within group(order by spots.km) as """median"""
FROM spots
WHERE
  $__timeFilter("time") AND
  rx_id = '$receiver_A' AND
  band = '$band_m_A'
GROUP BY 1
ORDER BY 1


SELECT
  $__timeGroupAlias("time",10m),
    percentile_disc(0.25) within group(order by spots.km) as """lower
quartile"""
FROM spots
WHERE
  $__timeFilter("time") AND
  rx_id = '$receiver_A' AND
  band = '$band_m_A'
GROUP BY 1
ORDER BY 1


SELECT
  $__timeGroupAlias("time",10m),
    percentile_disc(0.75) within group(order by spots.km) as """upper
quartile"""
FROM spots
WHERE
  $__timeFilter("time") AND
  rx_id = '$receiver_A' AND
  band = '$band_m_A'
GROUP BY 1
ORDER BY 1
```

An example Community HF Distance Stats dashboard is shown below, where receiver_A has been selected as WA2ZKD (New York State) and receiver_B as AI6VN/KH6 on Maui, and both bands are 20m - an example of where we want to compare the statistics for two different stations on the same band, here over 48 hours. The lower plot shows an example where we compare the statistics for the same station on two bands, here 20 and 40m.

Note that variables - used for the pull-down selections - are defined for a Dashboard, not individual panels. Therefore, the list is above Panel A, while Panel A only uses the first two variables and Panel B the second two.



wsprdaemon.org:3000/d/-0ONFs6Wz/community-hf-distance-stats?orgId=1

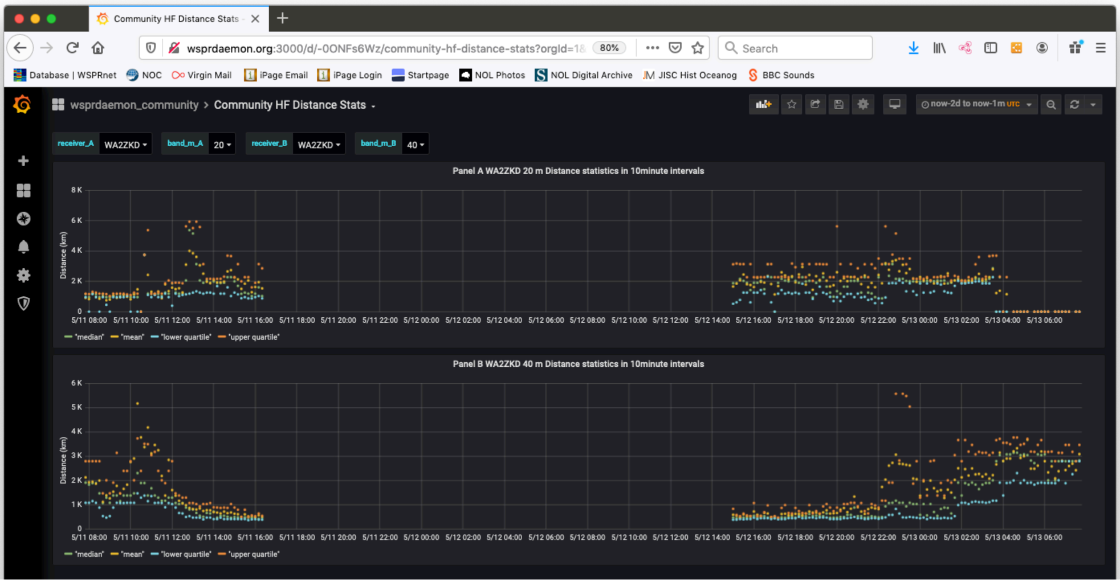
wsprdaemon_community › Community HF Distance Stats

receiver_A WA2ZKD band_m_A 20 receiver_B AI6VN/KH6 band_m_B 20

now-2d to now-1m UTC

Panel A WA2ZKD 20 m Distance statistics in 10minute intervals

Panel B AI6VN/KH6 20 m Distance statistics in 10minute intervals



wsprdaemon.org:3000/d/-0ONFs6Wz/community-hf-distance-stats?orgId=1

wsprdaemon_community › Community HF Distance Stats

receiver_A WA2ZKD band_m_A 20 receiver_B WA2ZKD band_m_B 40

now-2d to now-1m UTC

Panel A WA2ZKD 20 m Distance statistics in 10minute intervals

Panel B WA2ZKD 40 m Distance statistics in 10minute intervals