Data Selection Process

Selecting Days with Rain
Rain data from the Davis weather station is examined to find days when it rained at the radar site. Any day with more than 1.0mm of rain is chosen for analysis.
The files are located here on the gws:
/gws/nopw/j04/ncas_obs/amf/raw_data/ncas-aws-2/incoming/raine/NOAA/

Processing the vertical radar scans

The function that does the following processing is *process_zdr_scans* in *utilities/calib_functions.py*

The top level scripts are in *calc_calib*:
process_vert_scans_day.py and *process_vert_scans_time_series.py*
For each day with rain over the radar, the vertical scans from the radar are examined.

1. RAIN-E vertical scans use a single PRF rather than dual, so the velocity range is narrow. This means that sometimes the data is folded in heavy rain. The unambiguous velocity is 8m/s. I've applied a very simple unfolding procedure that seems to do the job:

```
ind = v>6.0
v2[ind] = -7.9725 - (7.9725-v[ind])
```

In rain, any large positive velocity (upwards) is likely to be because the downward velocity has folded. This procedure looks for any velocity greater than +6 and unfolds it. For example, a velocity of +7m/s is unfolded to -9m/s.

2. Next step is to calculate a simple average profile of Z, RhoHV, ZDR and V over the 360 degrees. I don't convert Z to linear units before I average, as it doesn't make a huge amount of difference.

3. I then look for data that meets the following criteria: rhv_prof > 0.99, uzh_prof > 0 and v_prof <- 2.
The starting point is the lowest range gate that meets these criteria.

4. Next is to look for gaps in this region of good data, as there could be multiple points from different parts of the profile. If there are more than 8 range dates between points then we select the top of the lowest range of data.

5. What is the depth of the data? Is it at least 0.5km? If so use it, otherwise discard the profile.

6. As we only want to be looking at raindrops, we need to make sure we are choosing data from below the melting layer. Next step is to look at the gradient of the velocity and RhoHV profiles to select the bottom of this layer, where there is a large gradient in these variables.

grad_v = np.gradient(v_prof)
grad_rhv = np.gradient(rhv_prof)

7. By examining multiple profiles, I came up with a threshold for grad_v and grad_rhv. The procedure has to find a large enough gradient of velocity within 0.5km of the region of extracted data (as described above). This prevents regions higher up in the profile being selected. The gradient in velocity alone is sufficient for the profile to be chosen, but if it is not quite large enough to meet the threshold but within a certain range, the gradient in RhoHV is then considered and if this is large enough, the profile can then be used.

if np.logical_or(np.logical_and(np.logical_and(max_grad_v_ind > ind3, 0.15 <= max_grad_v < 0.25), min_grad_rhv <-0.015), np.logical_and(max_grad_v_ind > ind3, max_grad_v >0.25)):

grad_v has to be greater than 0.25
if grad_v is between 0.15 and 0.25, then look at RhoHV, if RhoHV gradient is less than -0.015 (larger in the negative) then the profile is usable.

8. Calculate the median value of ZDR from the extracted profile.

9. The data for each day is saved in a file called day_ml_zdr.csv. Melting layer heights are derived, as being the height of the top of the range of the extracted data. This is the height of the ***bottom*** of the melting layer. This height can then be used in the Z calibration (although potentially it doesn't improve/help it.)

10. A separate function processes the daily files to calculate hourly values. There are about 9 radar scans each hour and there has to be at least 3 profiles in each hour to calculate an average hourly value. This is to exclude any values being derived from bad profiles/outliers that got through the above processing/thresholding.

process_hourly_zdr.py is used which calls the function *calc_hourly_ML* in *calib_functions.py*

ZDR bias from horizontal scans

We can also analyse ZDR from the low elevation PPI scans.
process_horz_zdr_day.py calls the function *horiz_zdr* from *calib_functions.py*

The script looks at low level scans, has a criteria for melting layer height taken from analysis of the vertical scans, and only looks below this level, and also thresholds on the following:

rhohv>0.99, phidp>0, phidp<6, uzh>15, uzh<=18

CSV files are created for each day

The script *plot_zdr_bias_time_series.py* is used to plot the results.

SCRIPT and OUTPUT SUMMARY

VERTICAL SCANS

STEP 1

- RUN process_vert_scans_time_series.py
    - CALLS process_vert_scans_day.py
        - CALLS process_zdr_scans in calib_functions.py in /gws/nopw/j04/ncas_obs/amf/software/ncas-mobile-x-band-radar-1/calc_calib

Produces the following files for each good rain profile:
day_ml_zdr.csv
vert_profs_20181031_2016.png
vert_profs_20181031_2016.txt

STEP 2

- RUN process_hourly_zdr.py
    - CALLS calc_hourly_ML in calib_functions.py

Produces hourly_ml_zdr.csv containing hourly values of ML and ZDR bias.

STEP 3

Run plot_zdr_bias_time_series.py to plot the results.

HORIZONTAL SCANS

STEP 1

- RUN process_horz_zdr_time_series.py
    - Calls process_horz_zdr_day.py
    - CALLS the function *horiz_zdr* from *calib_functions.py*

OUTPUT:
YYYYMMDD_horz_zdr.csv

STEP 2

- Run plot_horz_zdr_bias_time_series.py to plot the results