Short instruction on how to create a pywatershed experiment:

This documentation serves as a quick description of the test case provided and how to run a DA experiment with coupled pywatershed and DART. There are 4 general steps to take, which are as follows:

1. Prepare a case study. We have provided a case study here, which contains the domain data, forcing or inputs to the pywatershed model, initial restart files to be used and observations that are going to be assimilated. We explain the content in more detail below.
2. Prepare a yaml file containing the experiment details: A coupled pywatershed and DART experiment is controlled using the options provided in a yaml file which controls how to compile the models, whether and how to perturb the model forcing, the model parameters, the initial conditions and what options to use for running the model and DART. Sample yaml files are provided and will be further explained below.
3. Set up and run the pywatershed and DART coupled: the codes and scripts needed are committed in the DART main repository, and we will provide step by step instructions on how to setup and run.
4. Postprocess the experiment outputs: After completion of the experiment, we would post process the DART outputs and display the results in forms of box plots, and time series. Below we provide an explanation of each step.

1. Prepare a Case Study:

The current test case is the available pywatershed DRB test case distributed with the pywatershed code. Current domain folder for testing is located: `case_study/domain/drb_conus404` and here is the content of each folder

- **DOAMIN:** This folder contains all the files from pywatershed drb test case plus a newly generated files called “parameters_dis_seg_app.nc”, this file has been generated by the script `/PATH/TO/pywatershed/DART/models/pywatershed/scripts/append_seg_data.py` that 1- reads the lat/lon of the river segments from a csv file and added it to the parameter_dis_seg.nc file since it was missing from the original file coming from USGS, so we added it since DART needs it. 2- At the same time, it also adds the tracing information used in the localization scheme.
- **FORCING:** input data for our simulations is actually the model output of a full physics run for the simulation period, so one needs to run pywatershed once to generate the forcing files for the DA experiment.
    - Script used to generate the forcing is `case_study/domain/drb_conus404/full_physics_run/01_nhm.ipynb`

    - Note in order to run the channel only, we want this option in the full physics run: `control.options["netcdf_output_var_names"] += ["gwres_flow_vol", "sroff_vol", "ssres_flow_vol"]`
- **RESTART:** Restart files are generated by running the pywatershed code.
    - Script used to generate the restart files: `case_study/domain/drb_conus404/full_physics_run/02_restart_files_outputting.ipynb`
    - Note that the end time must come before the start date, otherwise the pywatershed code will crash
- **TimeSlices:** The observation convertor of the pywatershed is similar to WRF-Hydro and based on the time slices, rather than using the USGS download. Script `` `case_study/domain/drb/create_time_slices.R` `` reads the list of the gauges and creates time slices which are used by the DART observation converter.
    - Note: the observations are in CMS but the model is in CFS. The DART code will take care of the conversion between CMS and CFS.

2. Prepare a Yaml Config File:

One needs a yaml file that is unique to the pywatershed model and unique to the experiment in hand. We have provided a sample yaml file as part of the case study which can be found here: `/case_study/domain/experiment_config_files/drb_80mem_outflow_ts.yaml`

Note, all the addresses need to be modified before using the yaml file. The yaml file has all the information on how to set up the DA experiment, and has the following sections which need to be carefully set up. Currently the yaml has the following sections:

- **dart:** this section is set up to compile DART for the experiment and copy the executables into the experiment directory.
- **pywatershed:** Path to the domain directory and modules to run the model with. Currently we are supporting only the groundwater and channel routing.
- **experiment:** Path to the experiment and run directory.
- **forcing_ensemble:** This section generated a forcing ensemble from deterministic simulations, the user would provide the list of the forcing variables to perturb and the min/max value for random generator.
- **ensemble:** creates an ensemble of the model parameters based on the number of members, and perturbing a subset of parameters that is listed in `params_to_perturb`.
- **initial_ens:** This section generates an ensemble of initial conditions based on a single restart file by running the filter.
- **observation_preparation:** This section creates DART obs_seq files based on the time slides, it will generate one file per day.
    - Note: if you would like to perform an open loop analysis, set the `obs_fraction_for_error` to a large number. In that case, the observation is so uncertain that it will be rejected.

- **Run_experiment:** in this section, user would set the setup options such as the end time, filter options, #of cores to execute filter and the model with and etc.

3. Setup and Run the PWS_DART:

Following steps needs to be followed:

1. Install pywatershed, by setting up the “pws” conda environment. Note, we are using a modified version of pws right now: https://github.com/jmccreight/pywatershed/tree/feat_restart

2. Install the pws_dart.

```
cd PATH/TO/DART/models/pywatershed/pws_dart
pip install -e .
```

3. Run the setup script.

```
cd PATH/TO/DART/models/pywatershed/scripts
python setup_filter_experiment.py
PATH/TO/case_study/experiment_config_files/drb_80mem_outflow_ts.yaml
```

4. Run the experiment. After the setup is finished, it provides guides on how to submit the job to the queue.

Post DA analysis:

1. Collect the output from DART output directory, required inputs:
    - Run directory
    - Number of cores to use

```
python PATH/TO/DART/models/pywatershed/pws_dart/pwsdart/core/collect_run_output.py
--run_dir /PATH/TO/RUN_DIRECTORY/ --n_cores 6 --drop_variables "None"
--spatial_indices "None"
```

2. Create the obs_diag_output.nc and obs_epoch_001.nc file.

```
/PATH/TO/DART/models/pywatershed/scripts/obs_space_diags.sh
/PATH/TO/EXPERIMENT_DIRECTORY /PATH/TO/DIAGNOSTIC_DIRECTORY 2020-01-01_00
2020-06-30_00
```

3. Run the matlab script for plotting and summarizing the results. The first script will summarize the statistics across the DA period. The second will generate the hydrographs for all gauge locations in the domain.

```
/PATH/TO/DART/models/pywatershed/matlab/summary_boxplots_pws.m
/PATH/TO/DART/models/pywatershed/matlab/run_pwsDARTdiags.m
```