

pestpp-ies

July 1, 2019

1 Run PESTPP-IES

```
In [1]: %matplotlib inline
import os
import shutil
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
import matplotlib as mpl
plt.rcParams['font.size']=12
import flopy
import pyemu
%matplotlib inline
```

flopy is installed in /Users/jeremyw/Dev/gw1876/activities_csiro/notebooks/flopy

1.1 SUPER IMPORTANT: SET HOW MANY PARALLEL WORKERS TO USE

```
In [2]: num_workers = 20
```

```
In [3]: t_d = "template"
m_d = "master_ies"
```

```
In [4]: pst = pyemu.Pst(os.path.join(t_d, "freyberg.pst"))
pst.write_par_summary_table(filename="none")
```

```
Out[4]:
```

		type	transform	count	initial value	upper bound	\
gr_prsity5	gr_prsity5	log	705	0	0.176091		
pp_vka1	pp_vka1	log	32	0	1		
cn_ss7	cn_ss7	log	1	0	1		
cn_strt6	cn_strt6	log	1	0	0.0211893		
cn_sy6	cn_sy6	log	1	0	0.243038		
gr_vka5	gr_vka5	log	705	0	1		
cn_vka8	cn_vka8	log	1	0	1		
gr_rech3	gr_rech3	log	705	0	0.0413927		
cn_sy7	cn_sy7	log	1	0	0.243038		
gr_sy3	gr_sy3	log	705	0	0.243038		

cn_sy8	cn_sy8	log	1	0	0.243038
gr_hk4	gr_hk4	log	705	0	1
strk	strk	log	40	0	2
pp_rech1	pp_rech1	log	32	0	0.0413927
cn_vka6	cn_vka6	log	1	0	1
pp_ss0	pp_ss0	log	32	0	1
pp_prsity2	pp_prsity2	log	32	0	0.176091
gr_ss5	gr_ss5	log	705	0	1
gr_strt5	gr_strt5	log	705	0	0.0211893
cn_strt7	cn_strt7	log	1	0	0.0211893
pp_hk1	pp_hk1	log	32	0	1
cn_hk7	cn_hk7	log	1	0	1
cn_vka7	cn_vka7	log	1	0	1
gr_ss3	gr_ss3	log	705	0	1
cn_hk6	cn_hk6	log	1	0	1
gr_strt3	gr_strt3	log	705	0	0.0211893
gr_ss4	gr_ss4	log	705	0	1
flow	flow	log	1	0	0.09691
pp_hk2	pp_hk2	log	32	0	1
gr_prsity3	gr_prsity3	log	705	0	0.176091
...
cn_strt8	cn_strt8	log	1	0	0.0211893
pp_sy0	pp_sy0	log	32	0	0.243038
cn_rech4	cn_rech4	log	1	0	0.0413927
pp_strt2	pp_strt2	log	32	0	0.0211893
cn_rech5	cn_rech5	log	1	0	0.0413927
gr_sy5	gr_sy5	log	705	0	0.243038
gr_vka4	gr_vka4	log	705	0	1
gr_rech2	gr_rech2	log	705	0	0.0413927
cn_prsity7	cn_prsity7	log	1	0	0.176091
gr_sy4	gr_sy4	log	705	0	0.243038
pp_prsity1	pp_prsity1	log	32	0	0.176091
drncond_k00	drncond_k00	log	10	0	1
cn_hk8	cn_hk8	log	1	0	1
pp_ss2	pp_ss2	log	32	0	1
pp_sy2	pp_sy2	log	32	0	0.243038
welflux_k02	welflux_k02	log	6	0	1
gr_strt4	gr_strt4	log	705	0	0.0211893
gr_vka3	gr_vka3	log	705	0	1
pp_strt0	pp_strt0	log	32	0	0.0211893
pp_strt1	pp_strt1	log	32	0	0.0211893
gr_hk5	gr_hk5	log	705	0	1
pp_sy1	pp_sy1	log	32	0	0.243038
pp_vka2	pp_vka2	log	32	0	1
pp_vka0	pp_vka0	log	32	0	1
welflux	welflux	log	2	0	1
gr_hk3	gr_hk3	log	705	0	1
cn_prsity8	cn_prsity8	log	1	0	0.176091

cn_ss8	cn_ss8	log	1	0	1
cn_ss6	cn_ss6	log	1	0	1
pp_prsity0	pp_prsity0	log	32	0	0.176091

	lower bound	standard deviation
gr_prsity5	-0.30103	0.11928
pp_vka1	-1	0.5
cn_ss7	-1	0.5
cn_strt6	-0.0222764	0.0108664
cn_sy6	-0.60206	0.211275
gr_vka5	-1	0.5
cn_vka8	-1	0.5
gr_rech3	-0.0457575	0.0217875
cn_sy7	-0.60206	0.211275
gr_sy3	-0.60206	0.211275
cn_sy8	-0.60206	0.211275
gr_hk4	-1	0.5
strk	-2	1
pp_rech1	-0.0457575	0.0217875
cn_vka6	-1	0.5
pp_ss0	-1	0.5
pp_prsity2	-0.30103	0.11928
gr_ss5	-1	0.5
gr_strt5	-0.0222764	0.0108664
cn_strt7	-0.0222764	0.0108664
pp_hk1	-1	0.5
cn_hk7	-1	0.5
cn_vka7	-1	0.5
gr_ss3	-1	0.5
cn_hk6	-1	0.5
gr_strt3	-0.0222764	0.0108664
gr_ss4	-1	0.5
flow	-0.124939	0.0554622
pp_hk2	-1	0.5
gr_prsity3	-0.30103	0.11928
...
cn_strt8	-0.0222764	0.0108664
pp_sy0	-0.60206	0.211275
cn_rech4	-0.0457575	0.0217875
pp_strt2	-0.0222764	0.0108664
cn_rech5	-0.0457575	0.0217875
gr_sy5	-0.60206	0.211275
gr_vka4	-1	0.5
gr_rech2	-0.0457575	0.0217875
cn_prsity7	-0.30103	0.11928
gr_sy4	-0.60206	0.211275
pp_prsity1	-0.30103	0.11928
drncond_k00	-1	0.5

cn_hk8	-1	0.5
pp_ss2	-1	0.5
pp_sy2	-0.60206	0.211275
welflux_k02	-1	0.5
gr_strt4	-0.0222764	0.0108664
gr_vka3	-1	0.5
pp_strt0	-0.0222764	0.0108664
pp_strt1	-0.0222764	0.0108664
gr_hk5	-1	0.5
pp_sy1	-0.60206	0.211275
pp_vka2	-1	0.5
pp_vka0	-1	0.5
welflux	-1	0.5
gr_hk3	-1	0.5
cn_prsity8	-0.30103	0.11928
cn_ss8	-1	0.5
cn_ss6	-1	0.5
pp_prsity0	-0.30103	0.11928

[65 rows x 7 columns]

1.1.1 Run PESTPP-IES in original mode and post process

```
In [5]: #pst.pestpp_options = {}
pst.pestpp_options["ies_num_reals"] = 75
pst.pestpp_options["ies_par_en"] = "prior.jcb"
pst.pestpp_options["ies_bad_phi_sigma"] = 1.75
pst.pestpp_options["overdue_giveup_fac"] = 10.0
pst.control_data.noptmax = 3
```

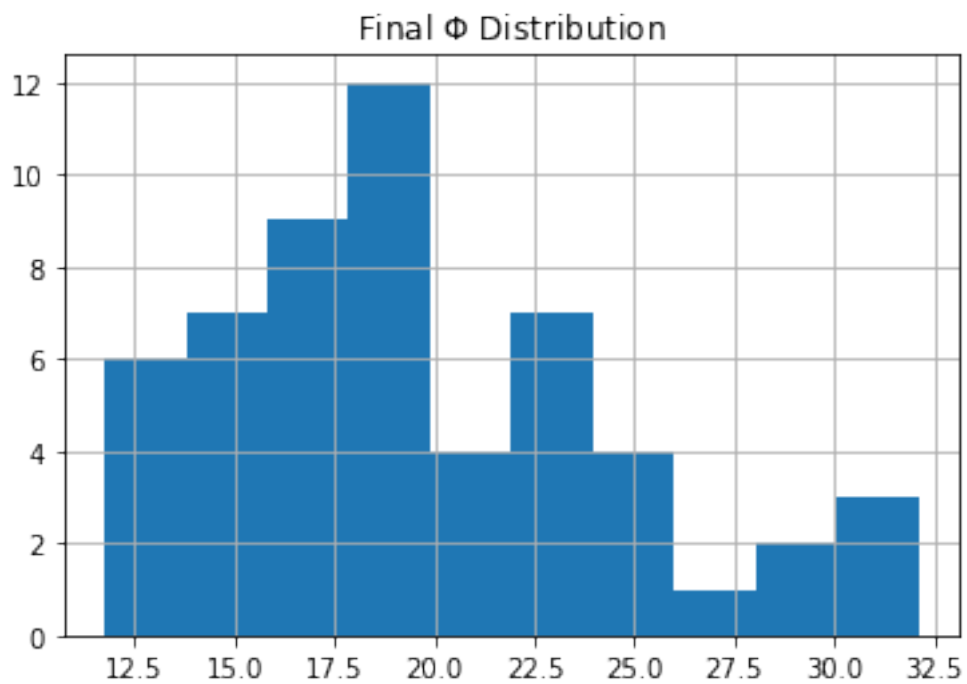
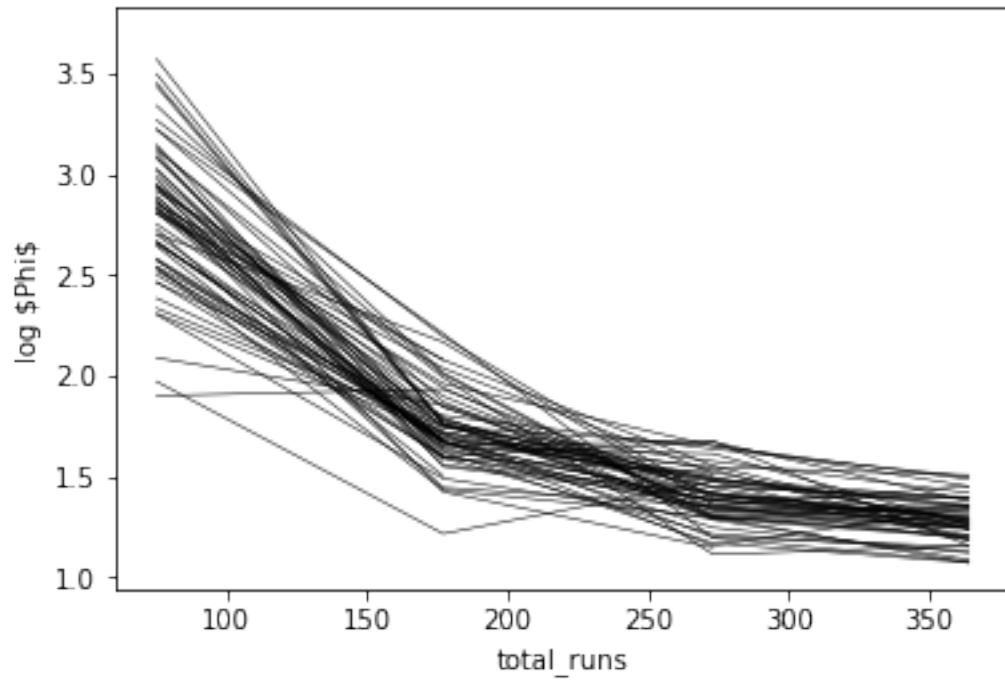
```
In [6]: pst.write(os.path.join(t_d, "freyberg_ies.pst"))
```

```
noptmax:3, npar_adj:14819, nnz_obs:14
```

```
In [7]: pyemu.os_utils.start_slaves(t_d, "pestpp-ies", "freyberg_ies.pst", num_slaves=num_workers)
```

A cheap phi progress plot

```
In [8]: phi = pd.read_csv(os.path.join(m_d, "freyberg_ies.phi.actual.csv"), index_col=0)
phi.index = phi.total_runs
phi.iloc[:, 6:].apply(np.log10).plot(legend=False, lw=0.5, color='k')
plt.ylabel('log \Phi')
plt.figure()
phi.iloc[-1, 6:].hist()
plt.title('Final \Phi Distribution');
```

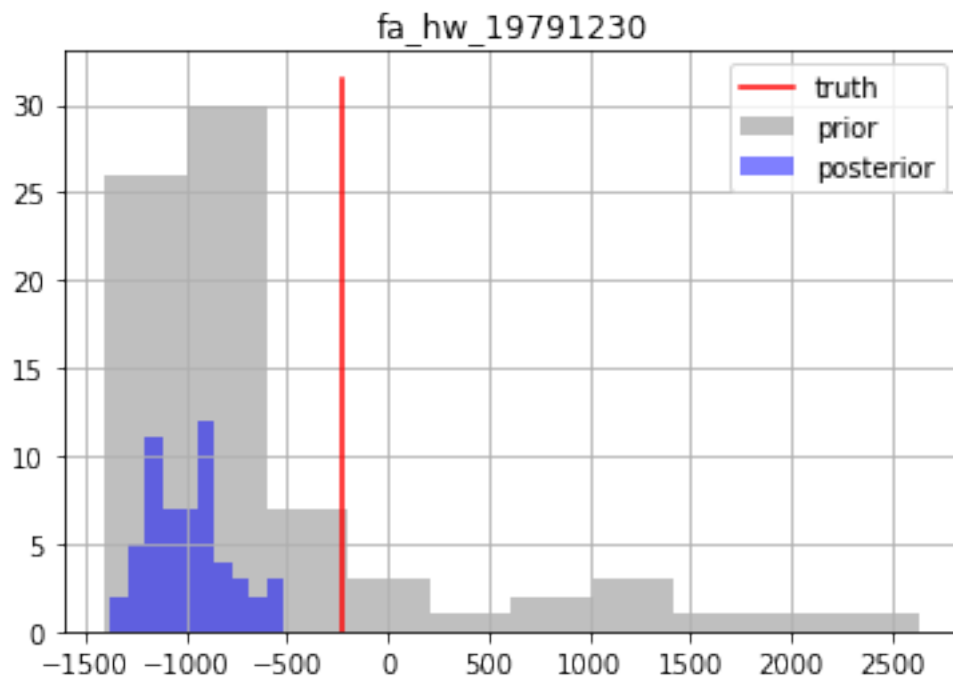


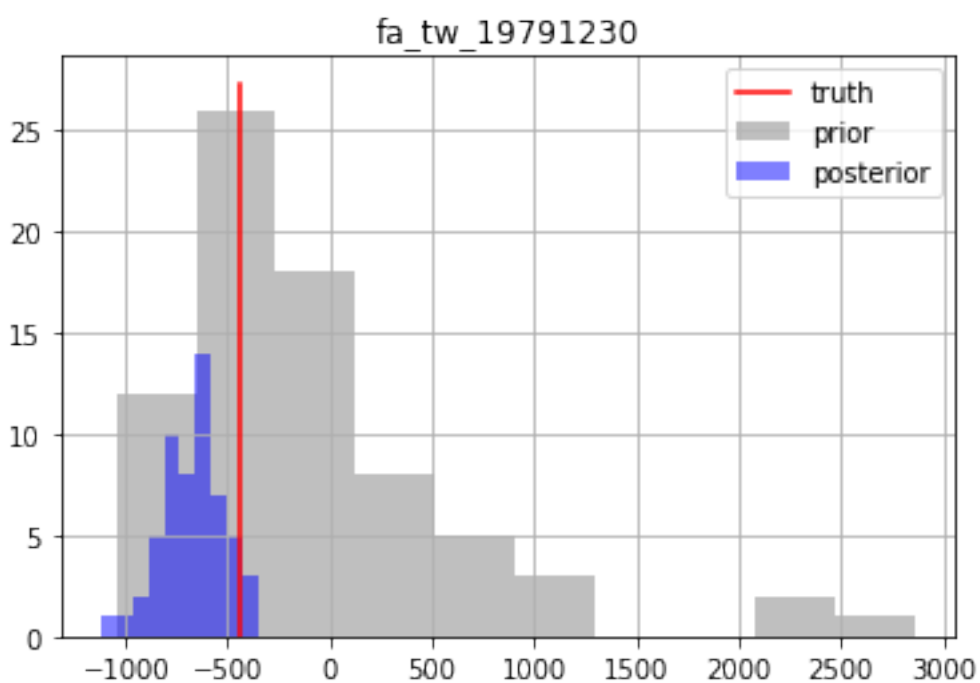
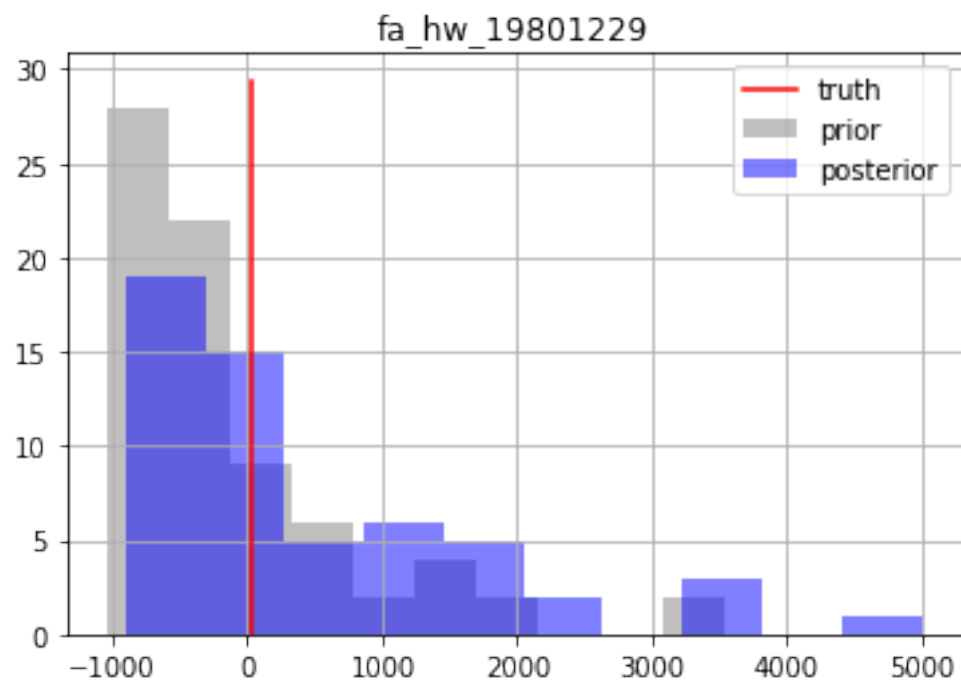
Plot forecast prior and posterior histograms with "truth" (red line)

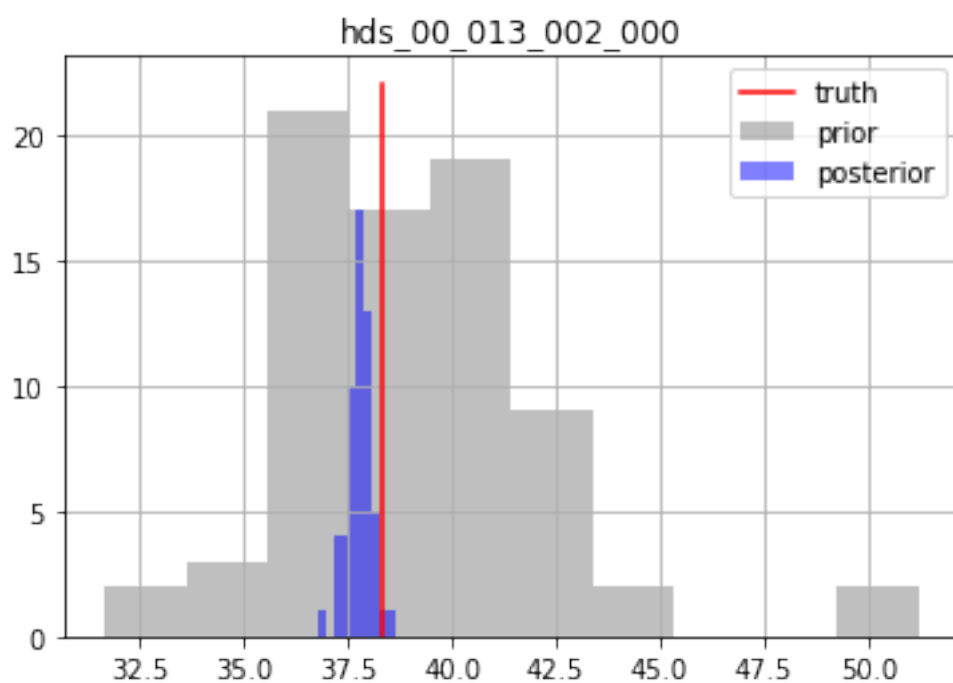
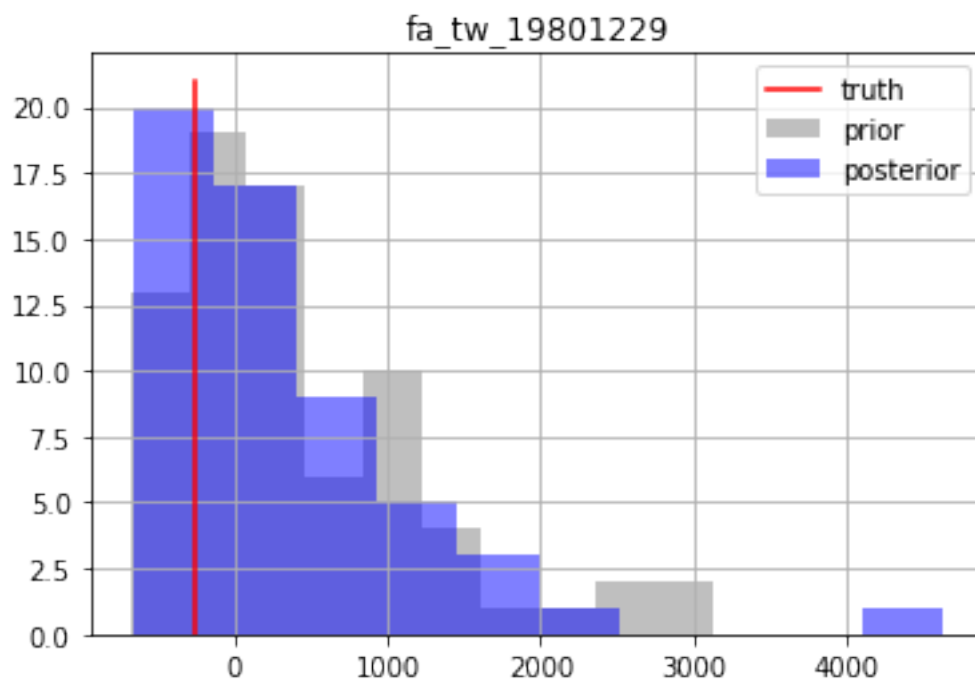
```

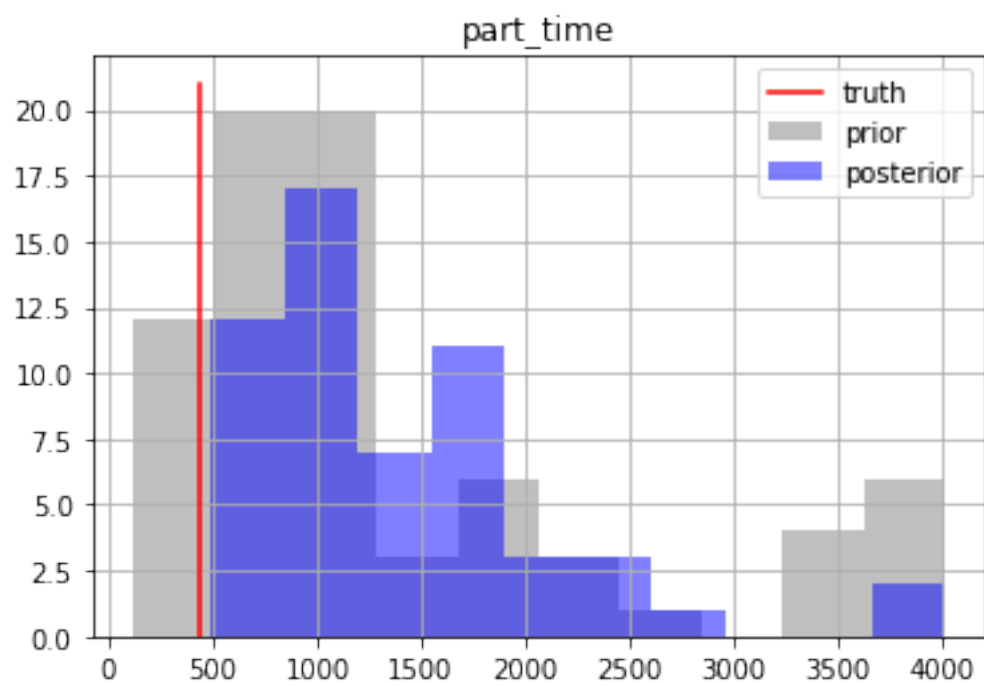
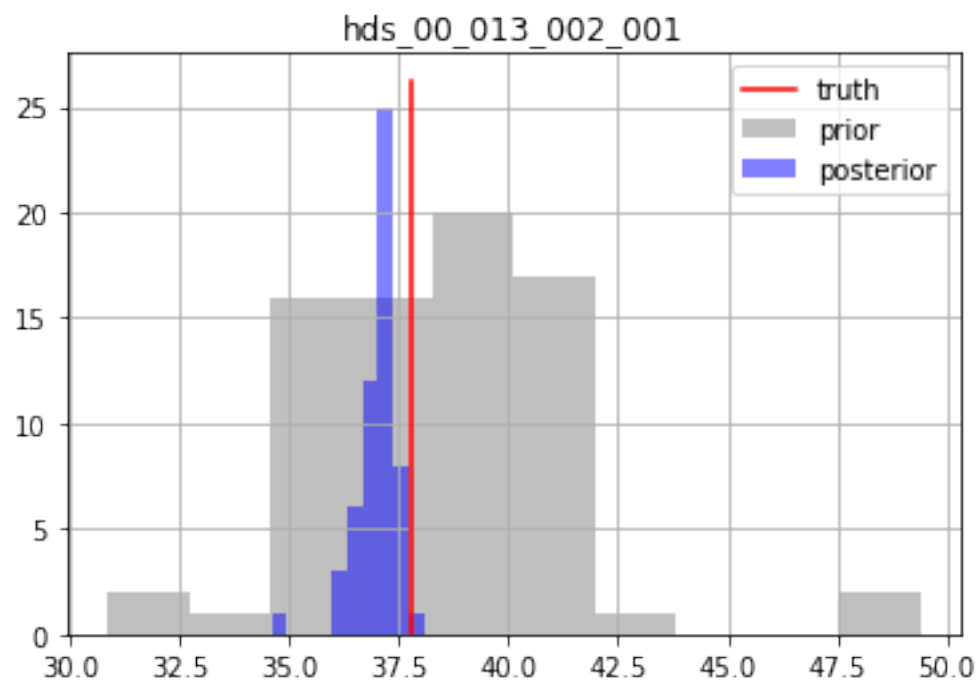
In [9]: oe_pr = pd.read_csv(os.path.join(m_d,"freyberg_ies.0.obs.csv"),index_col=0)
        oe_pt = pd.read_csv(os.path.join(m_d,"freyberg_ies.{0}.obs.csv".format(pst.control_data
        obs = pst.observation_data
        fnames = pst.pestpp_options["forecasts"].split(",")
        for forecast in fnames:
            ax = plt.subplot(111)
            oe_pr.loc[:,forecast].hist(ax=ax,color="0.5",alpha=0.5, label='prior')
            oe_pt.loc[:,forecast].hist(ax=ax,color="b",alpha=0.5, label='posterior')
            ax.plot([obs.loc[forecast,"obsval"],obs.loc[forecast,"obsval"]],ax.get_ylim(),"r",
            ax.set_title(forecast)
            ax.legend(loc='upper right')
            plt.show()

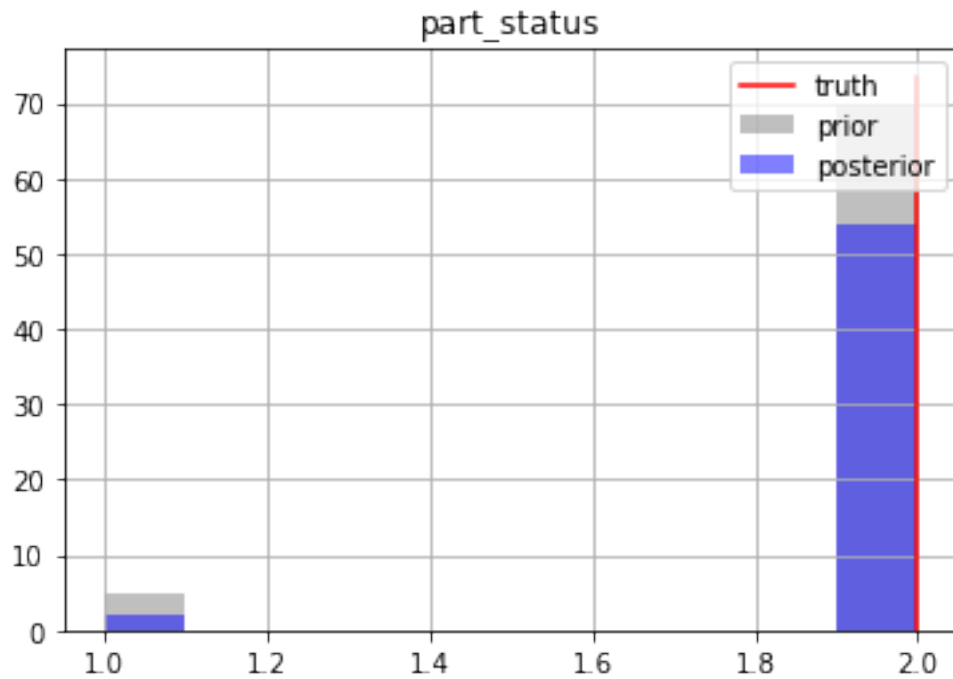
```











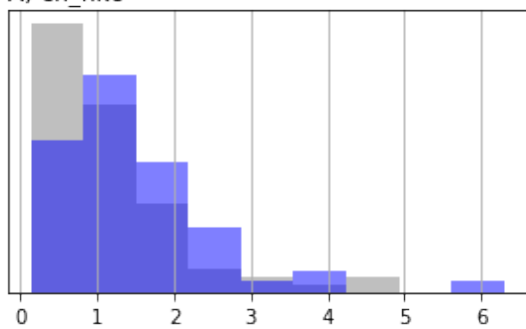
Plot parameter histograms by group

```
In [10]: pe_pr = pd.read_csv(os.path.join(m_d,"freyberg_ies.0.par.csv"),index_col=0)
         pe_pt = pd.read_csv(os.path.join(m_d,"freyberg_ies.{0}.par.csv".format(pst.control_da
         par = pst.parameter_data
         pdict = par.groupby("pargp").groups
         pyemu.plot_utils.ensemble_helper({"0.5":pe_pr,"b":pe_pt},plot_cols=pdict)

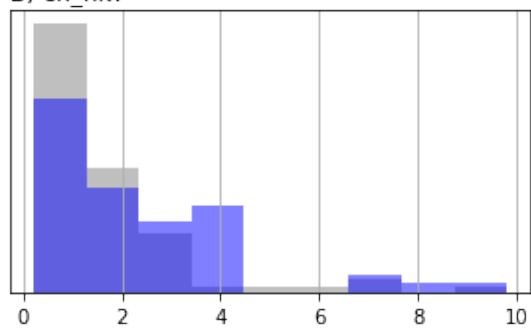
/Users/jeremyw/miniconda3/lib/python3.5/site-packages/IPython/core/interactiveshell.py:2785: D
interactivity=interactivity, compiler=compiler, result=result)
```

<Figure size 576x756 with 0 Axes>

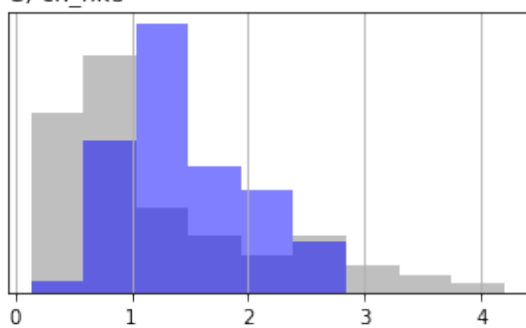
A) cn_hk6



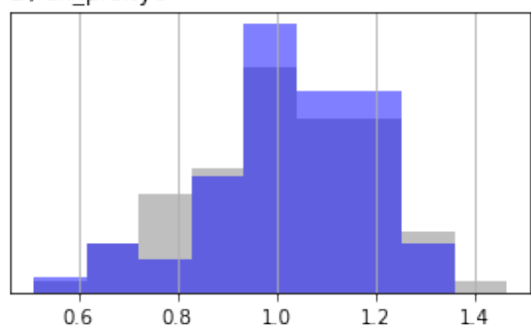
B) cn_hk7



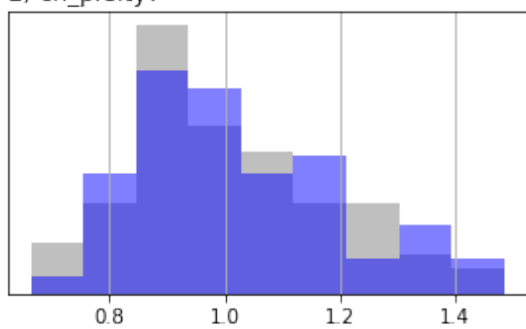
C) cn_hk8



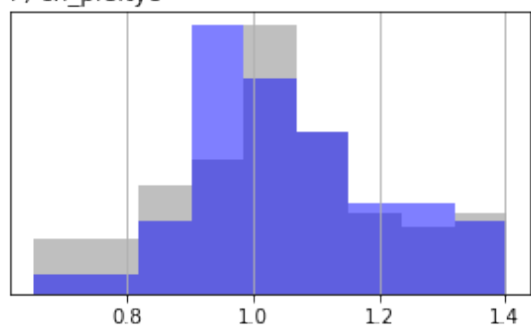
D) cn_prsity6



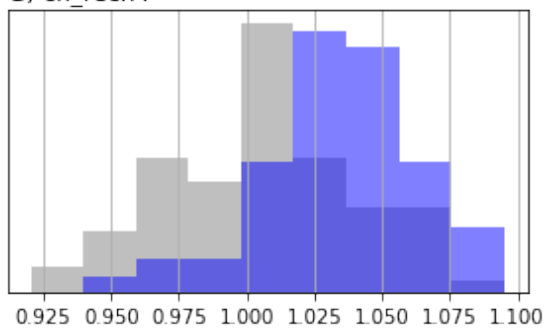
E) cn_prsity7



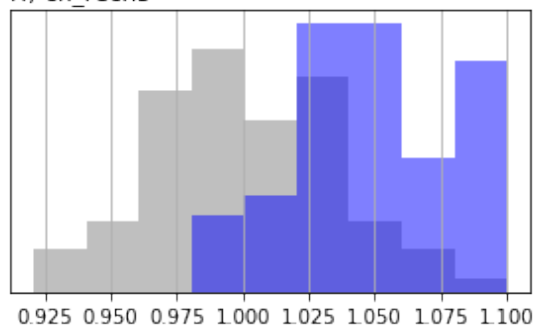
F) cn_prsity8

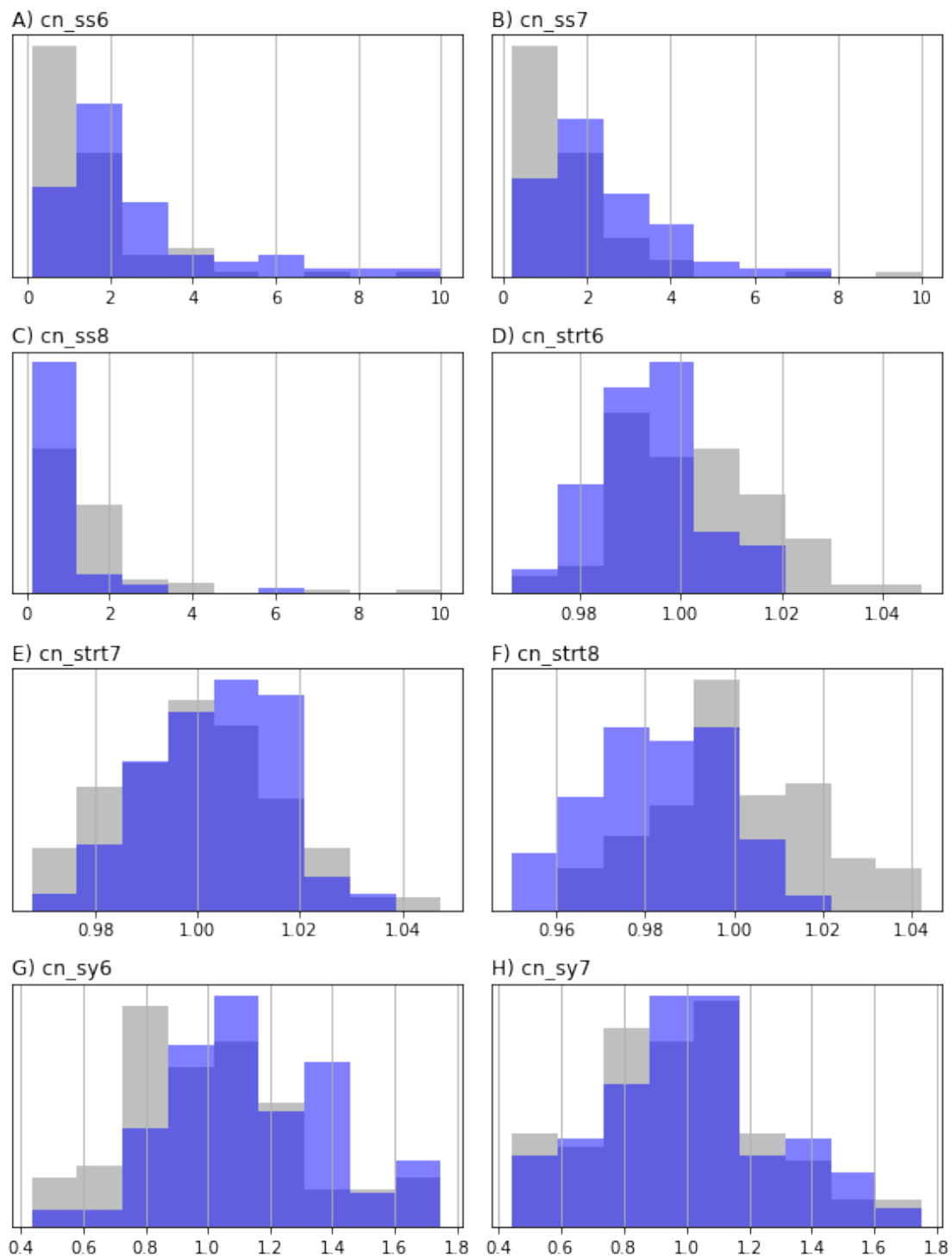


G) cn_rech4

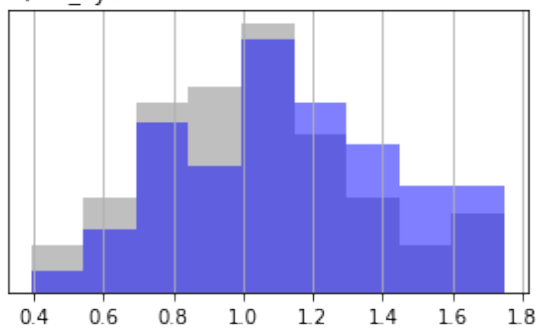


H) cn_rech5

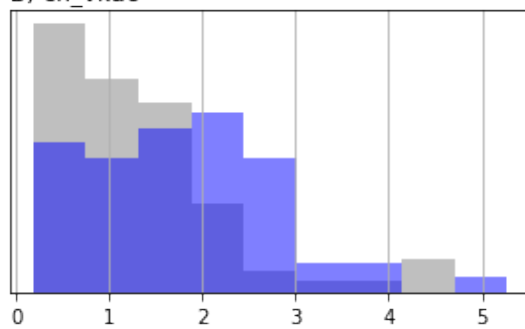




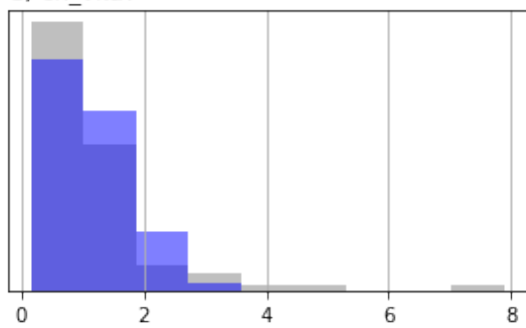
A) cn_sy8



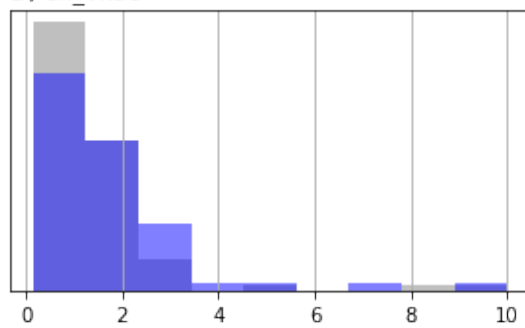
B) cn_vka6



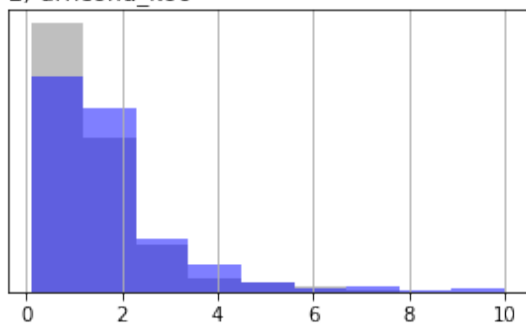
C) cn_vka7



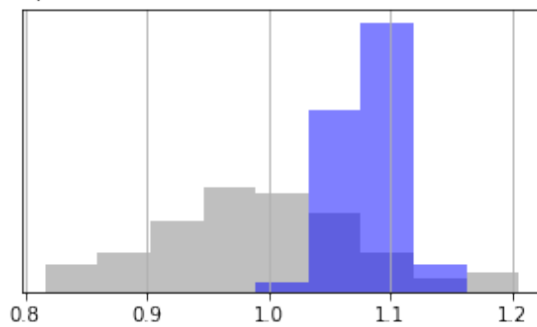
D) cn_vka8



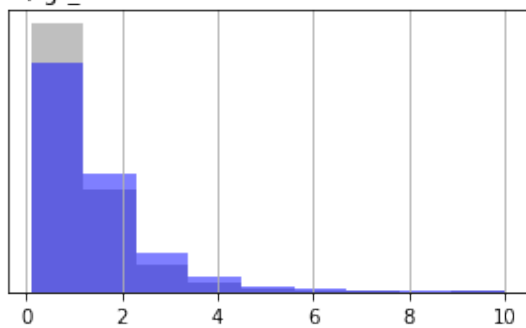
E) drncond_k00



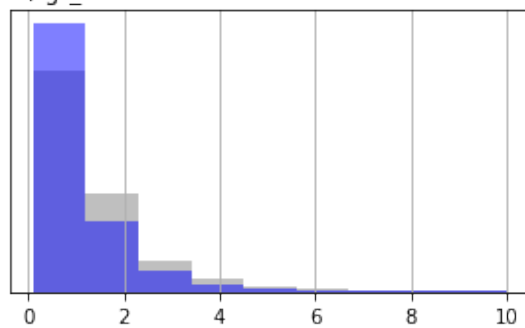
F) flow



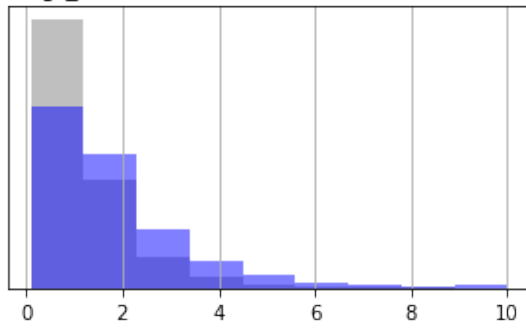
G) gr_hk3



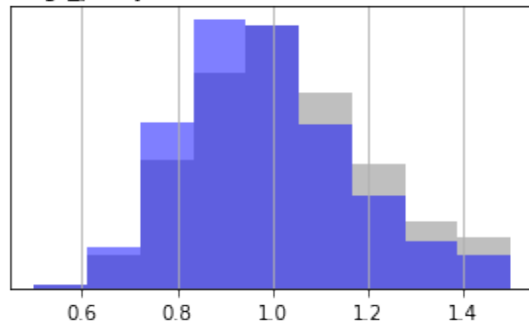
H) gr_hk4



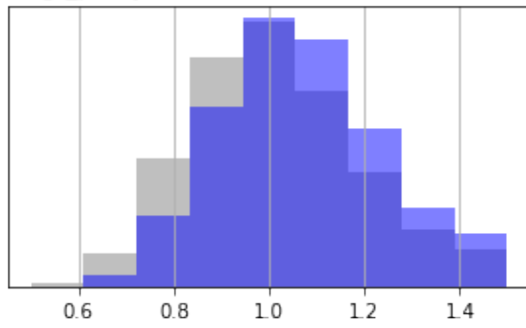
A) gr_hk5



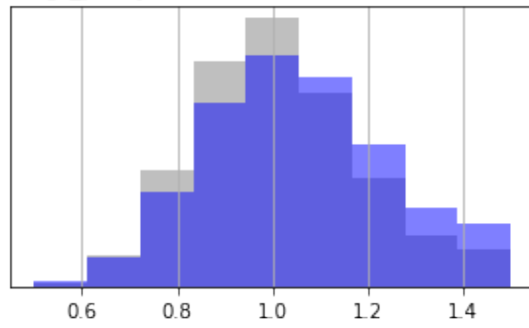
B) gr_prsity3



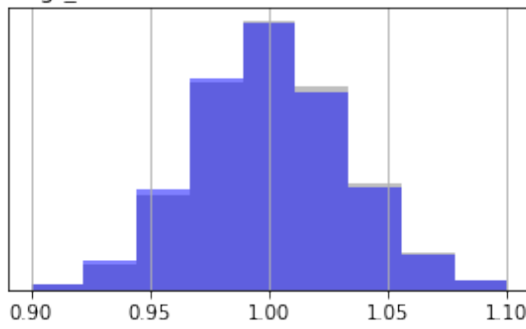
C) gr_prsity4



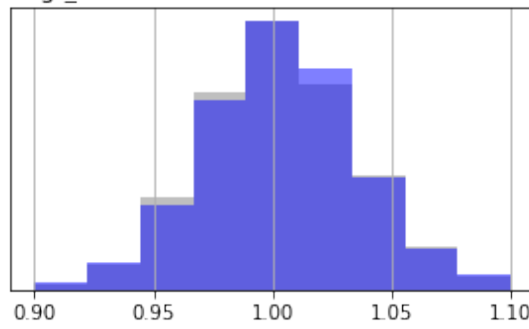
D) gr_prsity5



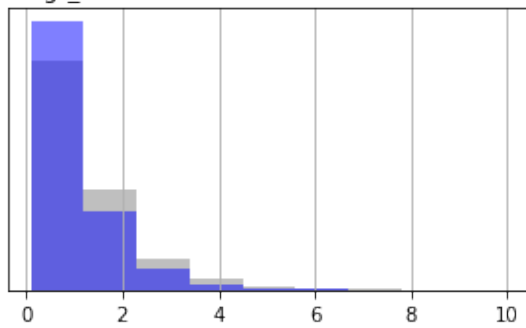
E) gr_rech2



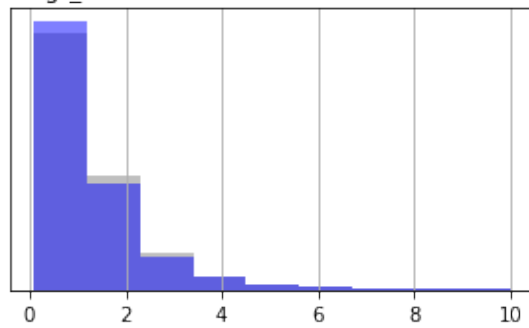
F) gr_rech3

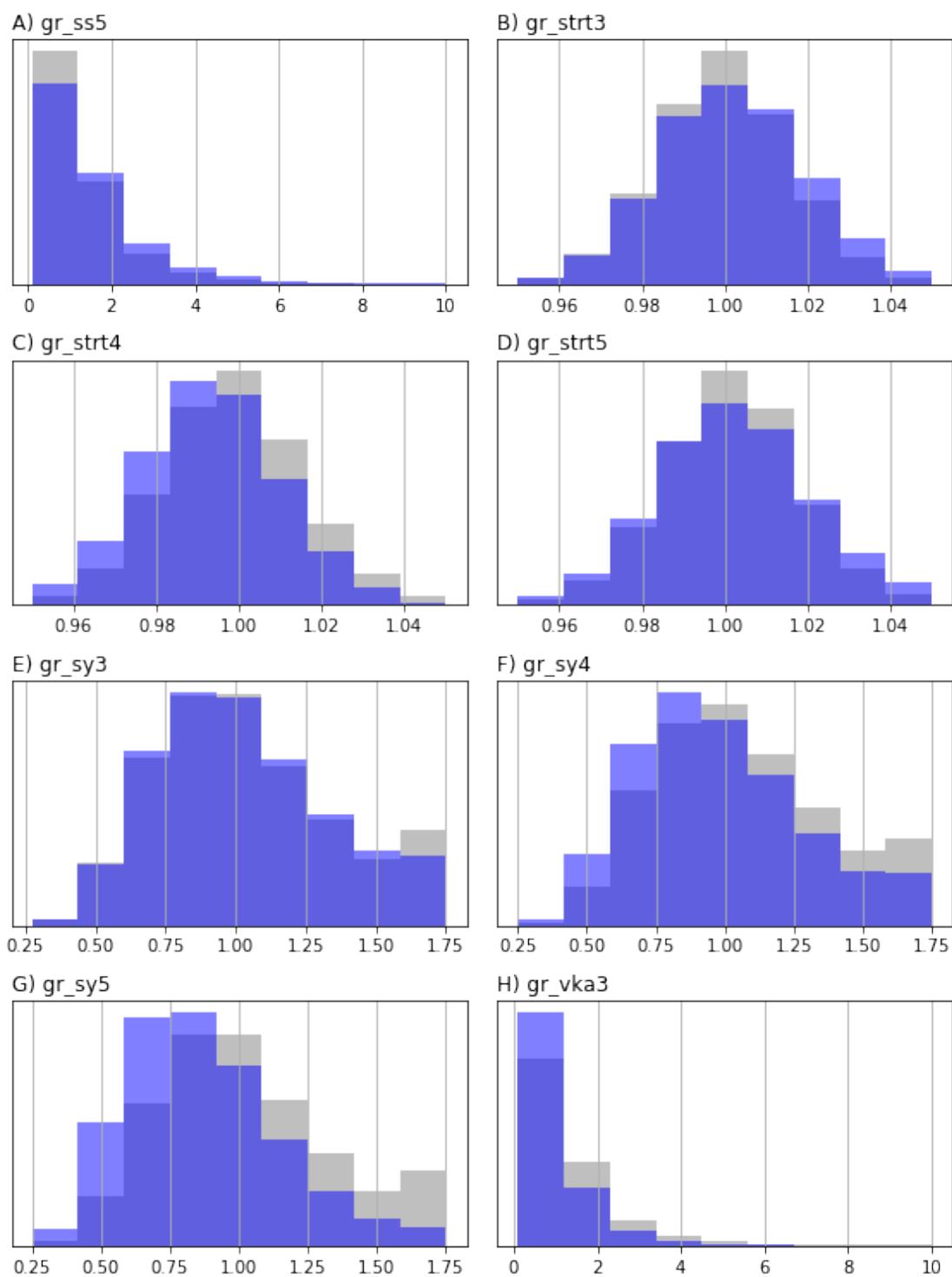


G) gr_ss3

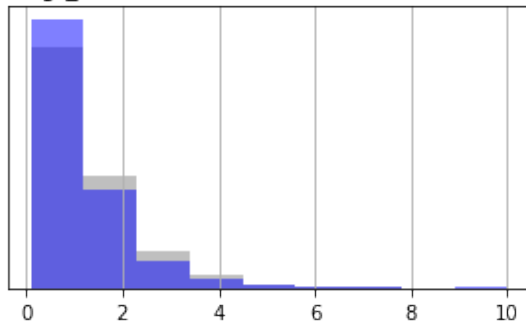


H) gr_ss4

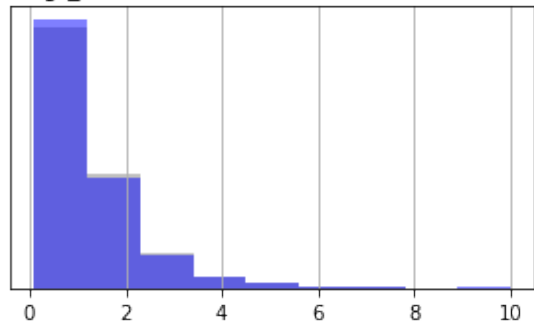




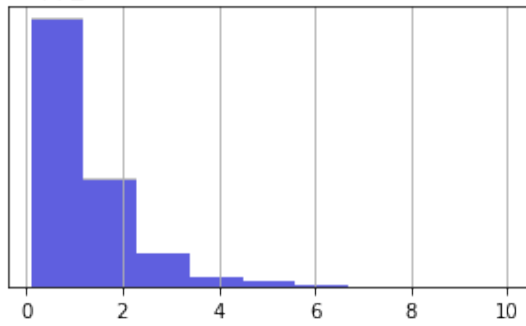
A) gr_vka4



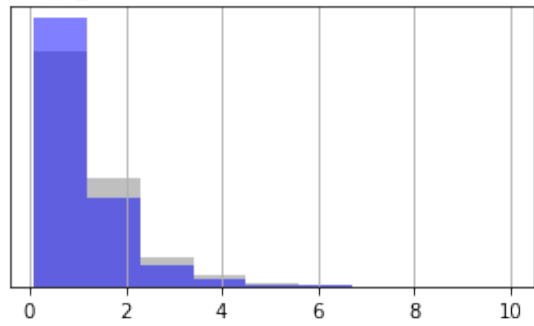
B) gr_vka5



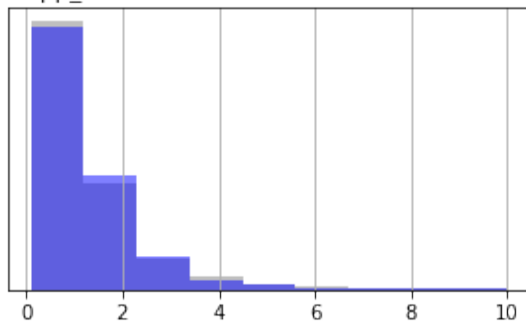
C) pp_hk0



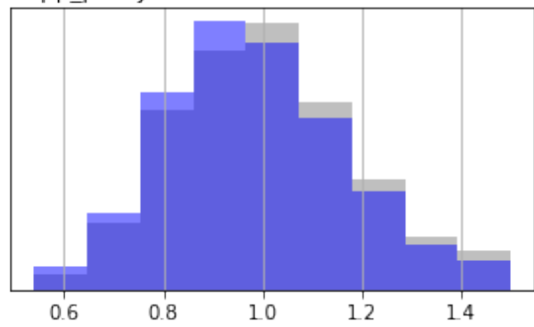
D) pp_hk1



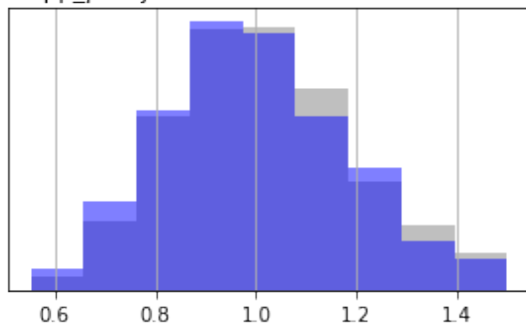
E) pp_hk2



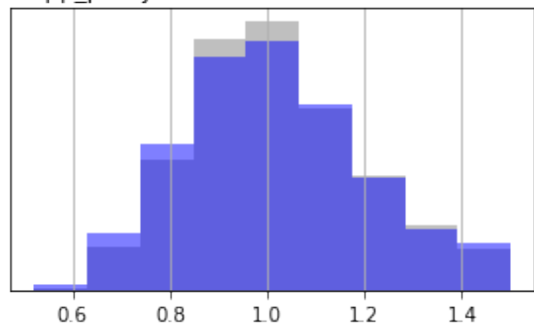
F) pp_prsity0



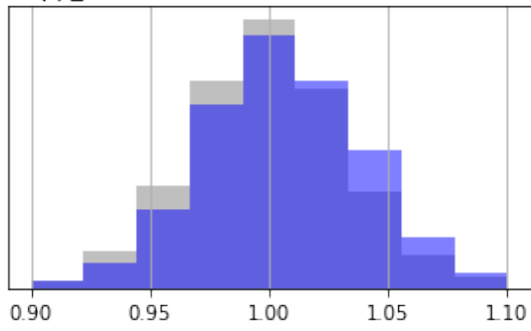
G) pp_prsity1



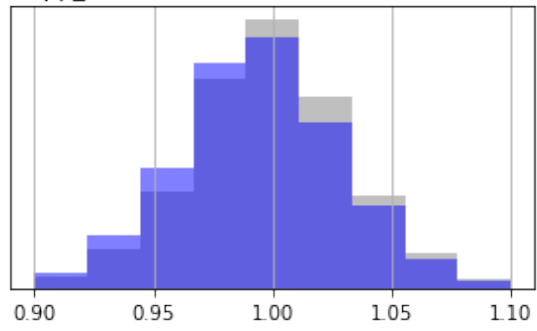
H) pp_prsity2



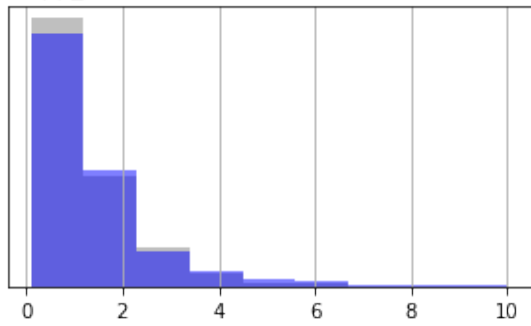
A) pp_rech0



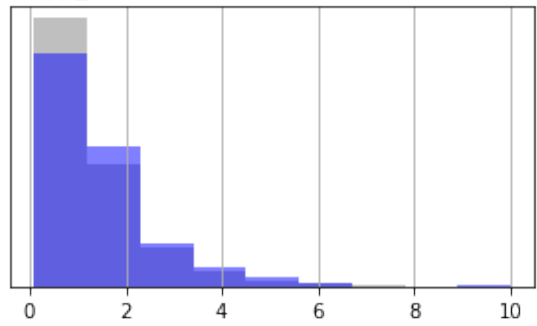
B) pp_rech1



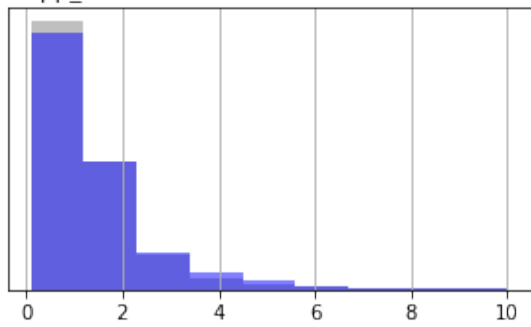
C) pp_ss0



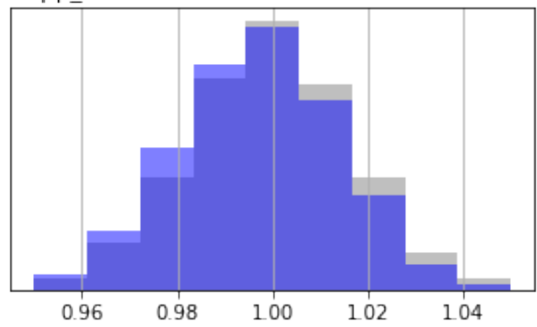
D) pp_ss1



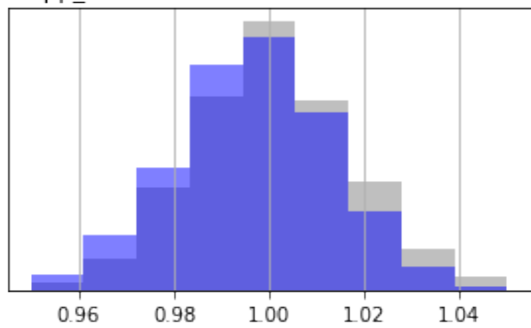
E) pp_ss2



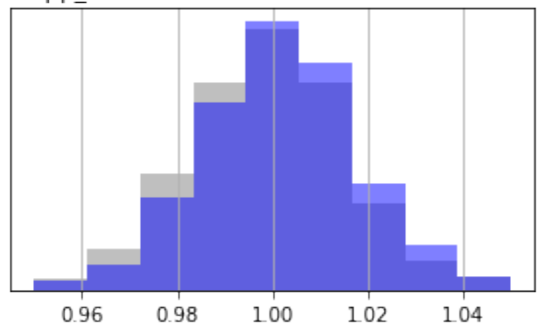
F) pp_strt0



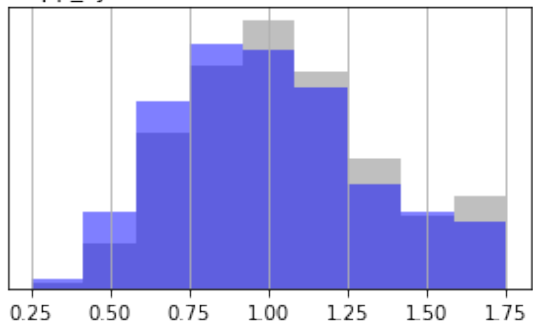
G) pp_strt1



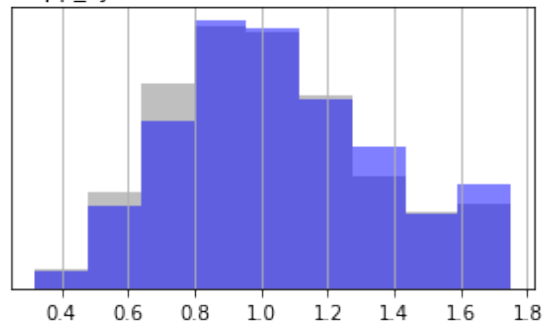
H) pp_strt2



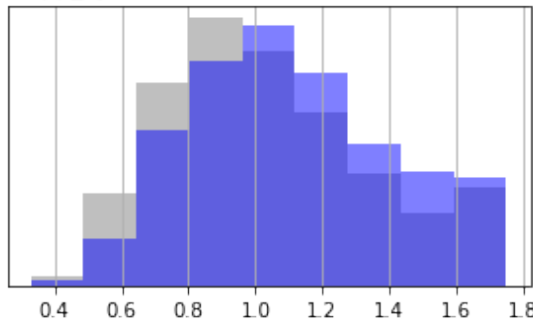
A) pp_sy0



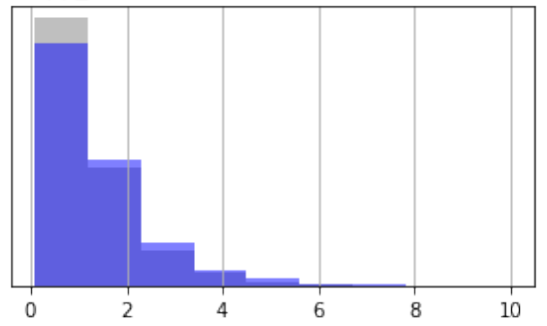
B) pp_sy1



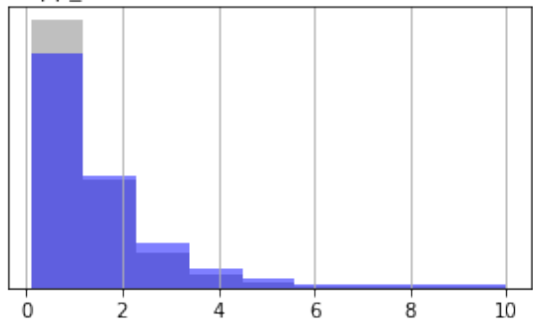
C) pp_sy2



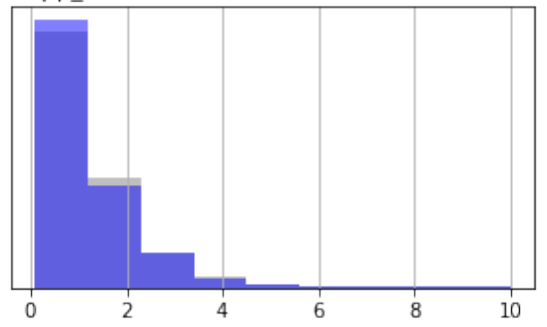
D) pp_vka0



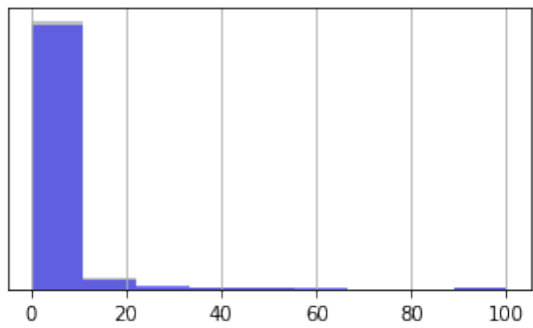
E) pp_vka1



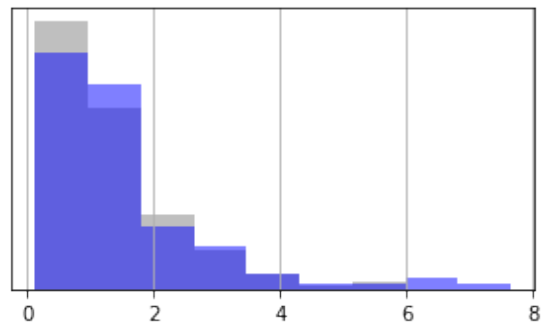
F) pp_vka2

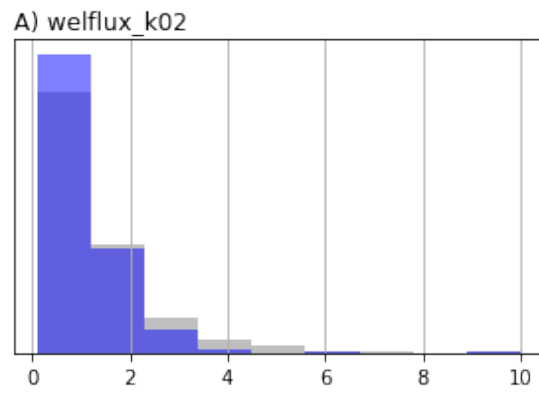


G) strk



H) welflux





```
In [11]: # pyemu.plot_utils.ensemble_change_summary(pe_pr,pe_pt,pst=pst,bins=20)
         # par = pst.parameter_data
         # li = par.partrans=="log"
```

```
# pe_pr.loc[:,li] = pe_pr.loc[:,li].apply(np.log10)
# pe_pr.shape
```

1.1.2 PESTPP-IES with simple temporal localization (and common sense)

Now let's add some localization. The obvious stuff is temporal - scenario parameters can't influence historic observations (and the inverse is true) so let's tell PESTPP-IES about this. Also, should porosity be adjusted at all given the observations we have???

```
In [12]: par = pst.parameter_data
         #parameter groups for future recharge
         dont_groups = [g for g in pst.par_groups if "pr" in g]
         dont_groups.extend(["gr_rech3", "pp_rech1", "cn_rech5"])
         dont_groups = [g for g in dont_groups if g in pst.adj_par_groups]
         dont_pars = par.loc[par.pargp.apply(lambda x: x in dont_groups), "parname"].tolist()
         dont_pars.append("welflux_001")
         dont_groups.append("welflux_001")
         dont_groups
```

```
Out[12]: ['gr_prsity5',
          'pp_prsity2',
          'gr_prsity3',
          'cn_prsity6',
          'gr_prsity4',
          'cn_prsity7',
          'pp_prsity1',
          'cn_prsity8',
          'pp_prsity0',
          'gr_rech3',
          'pp_rech1',
          'cn_rech5',
          'welflux_001']
```

```
In [13]: cols = pst.adj_par_groups
         cols.remove("welflux")
         cols.extend(["welflux_000", "welflux_001"])
         loc = pyemu.Matrix.from_names(pst.nnz_obs_names, cols).to_dataframe()
         loc.loc[:, :] = 1.0
         loc.loc[:, dont_groups] = 0.0
         pyemu.Matrix.from_dataframe(loc).to_ascii(os.path.join(t_d, "loc.mat"))
```

```
In [14]: pst.pestpp_options["ies_localizer"] = "loc.mat"
         pst.write(os.path.join(t_d, "freyberg_ies.pst"))
         pyemu.os_utils.start_slaves(t_d, "pestpp-ies", "freyberg_ies.pst", num_slaves=num_workers)
```

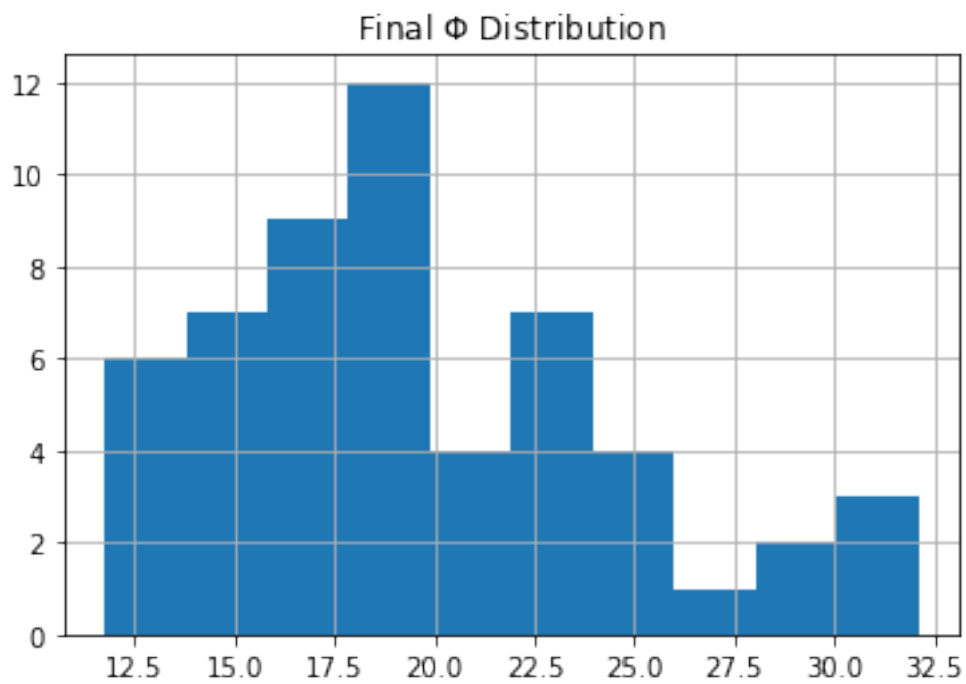
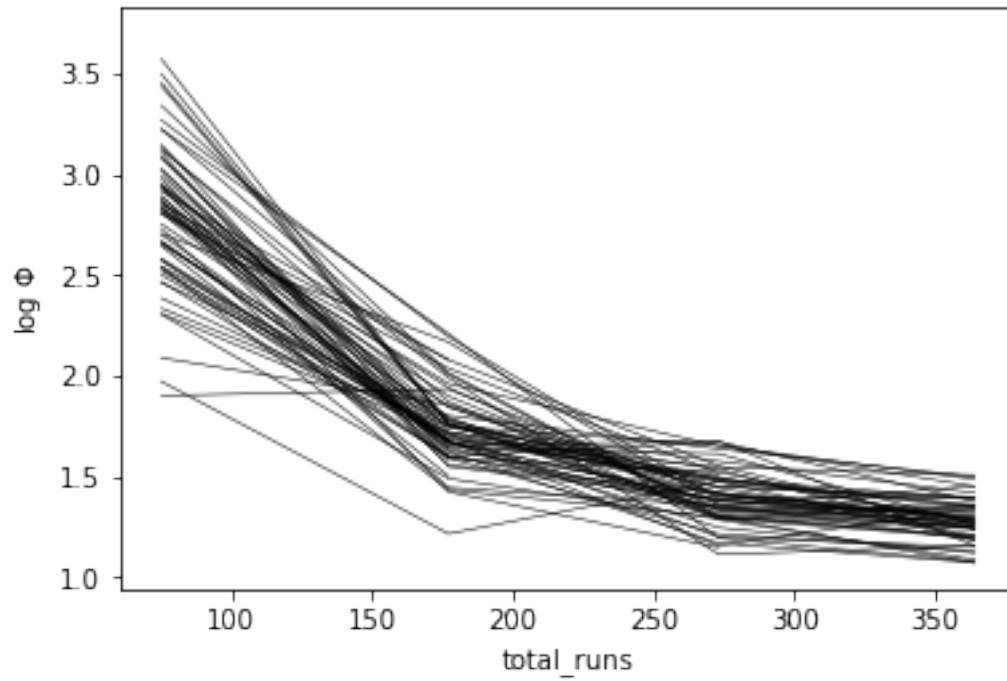
```
noptmax:3, npar_adj:14819, nnz_obs:14
```

```
In [15]: phi = pd.read_csv(os.path.join(m_d, "freyberg_ies.phi.actual.csv"), index_col=0)
         phi.index = phi.total_runs
```

```

phi.iloc[:,6:].apply(np.log10).plot(legend=False,lw=0.5,color='k')
plt.ylabel('log  $\Phi$ ')
plt.show()
phi.iloc[-1,6:].hist()
plt.title('Final  $\Phi$  Distribution');

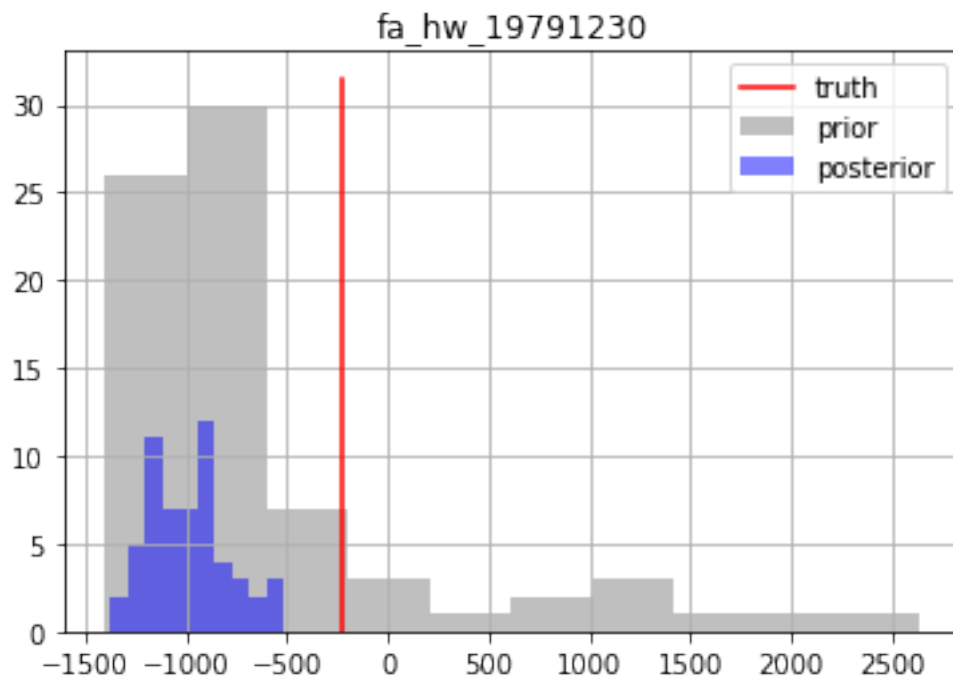
```

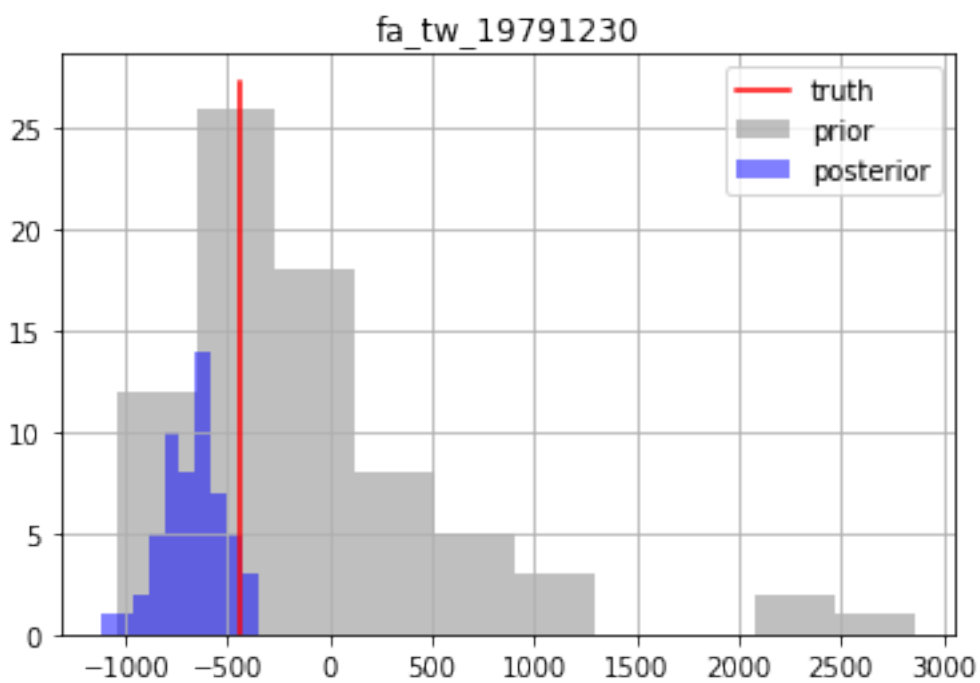
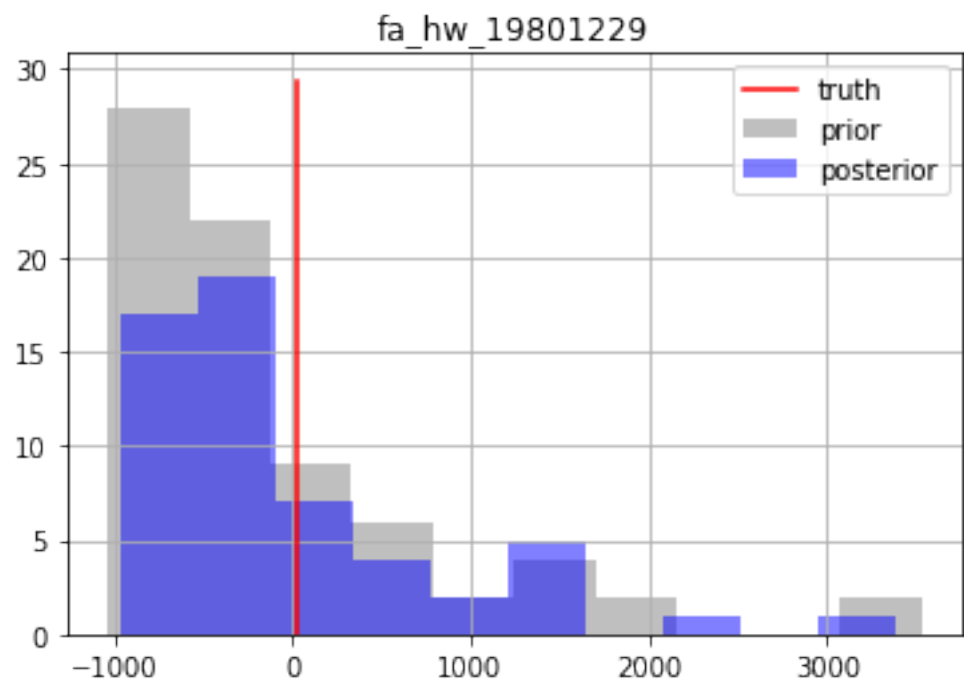


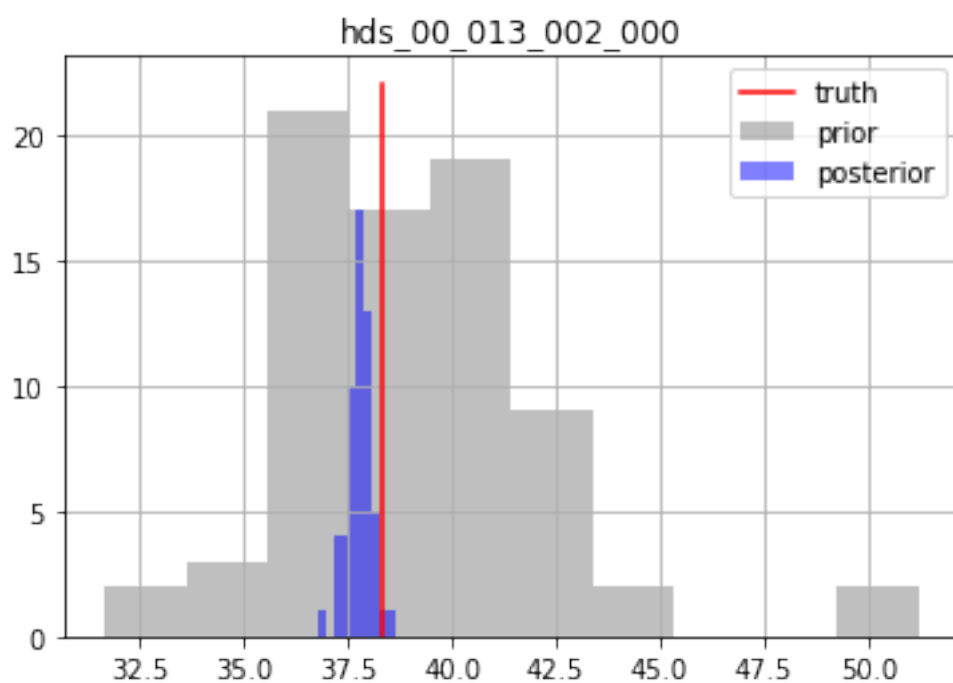
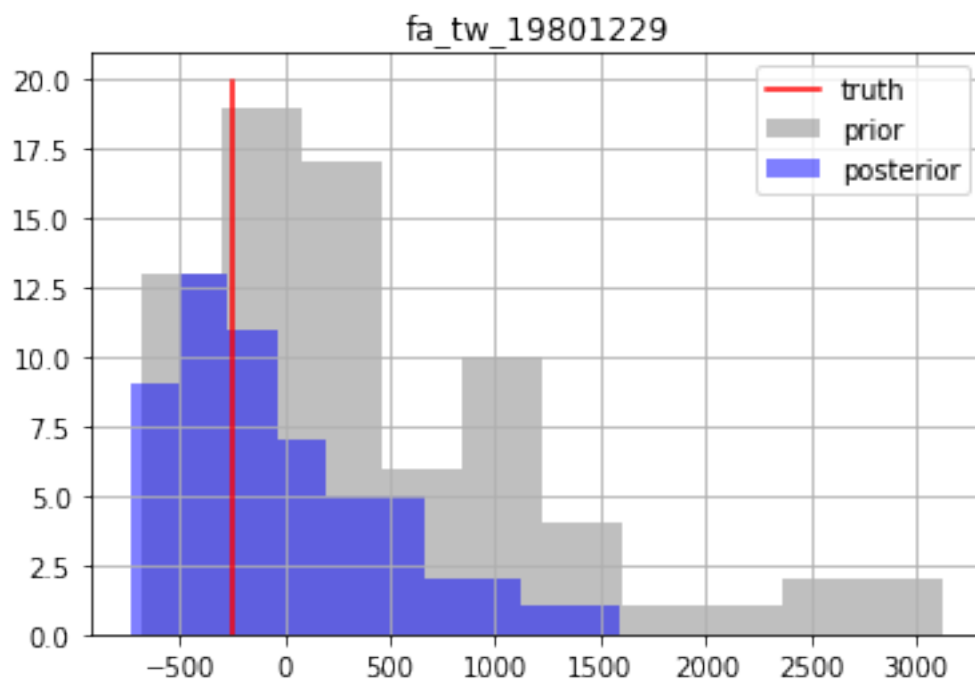
```

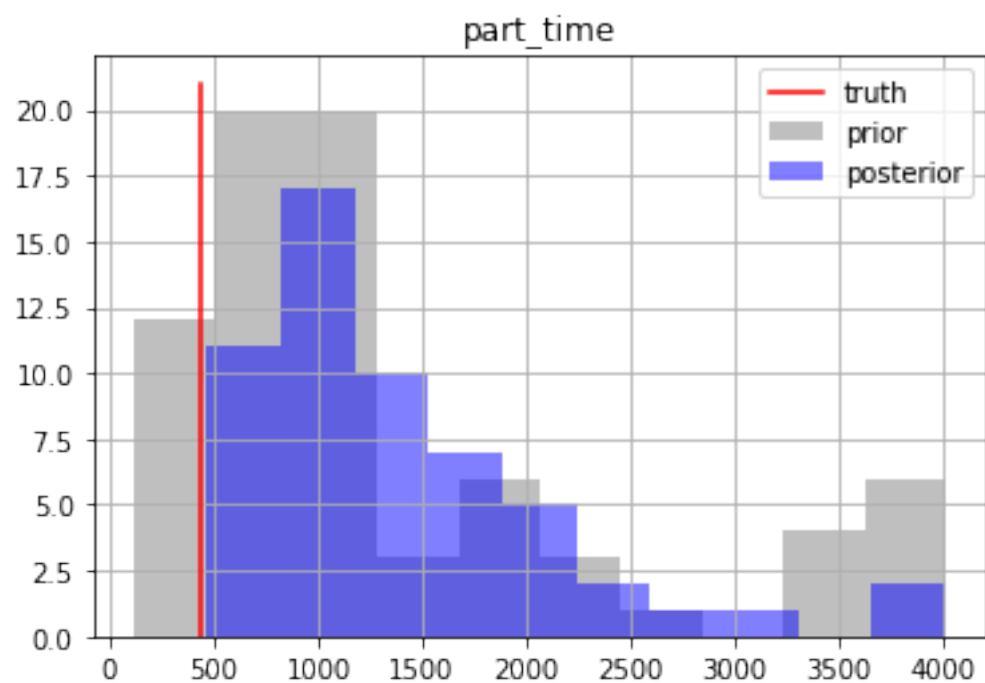
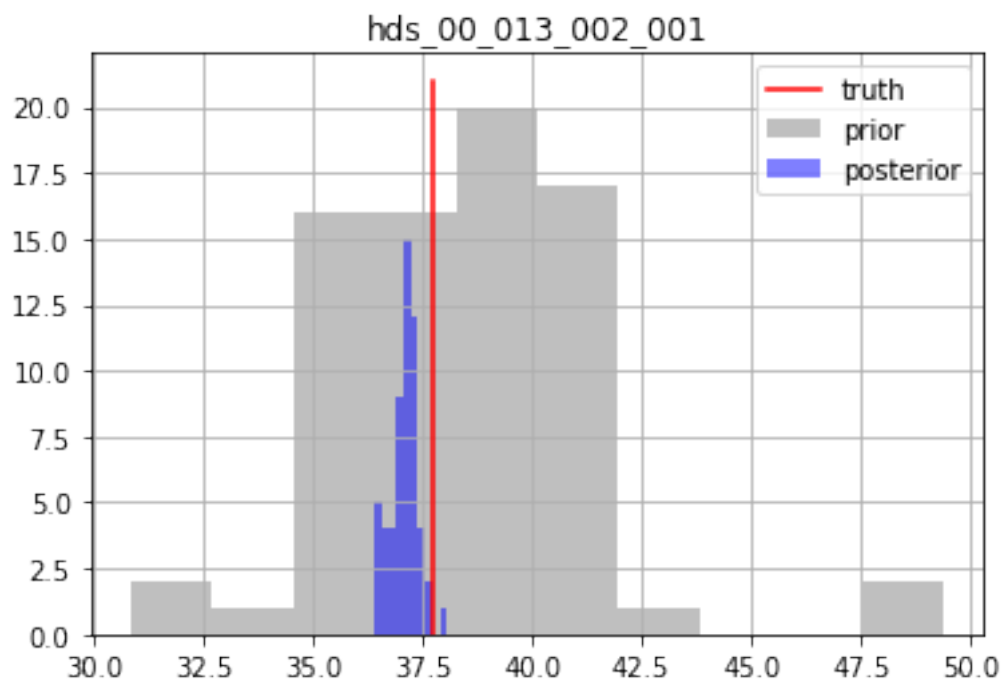
In [16]: oe_pr = pd.read_csv(os.path.join(m_d,"freyberg_ies.0.obs.csv"),index_col=0)
oe_pt = pd.read_csv(os.path.join(m_d,"freyberg_ies.{0}.obs.csv".format(pst.control_da
obs = pst.observation_data
fnames = pst.pestpp_options["forecasts"].split(",")
for forecast in fnames:
    ax = plt.subplot(111)
    oe_pr.loc[:,forecast].hist(ax=ax,color="0.5",alpha=0.5, label='prior')
    oe_pt.loc[:,forecast].hist(ax=ax,color="b",alpha=0.5, label='posterior')
    ax.plot([obs.loc[forecast,"obsval"],obs.loc[forecast,"obsval"]],ax.get_ylim(),"r")
    ax.set_title(forecast)
    ax.legend(loc='upper right')
    plt.show()

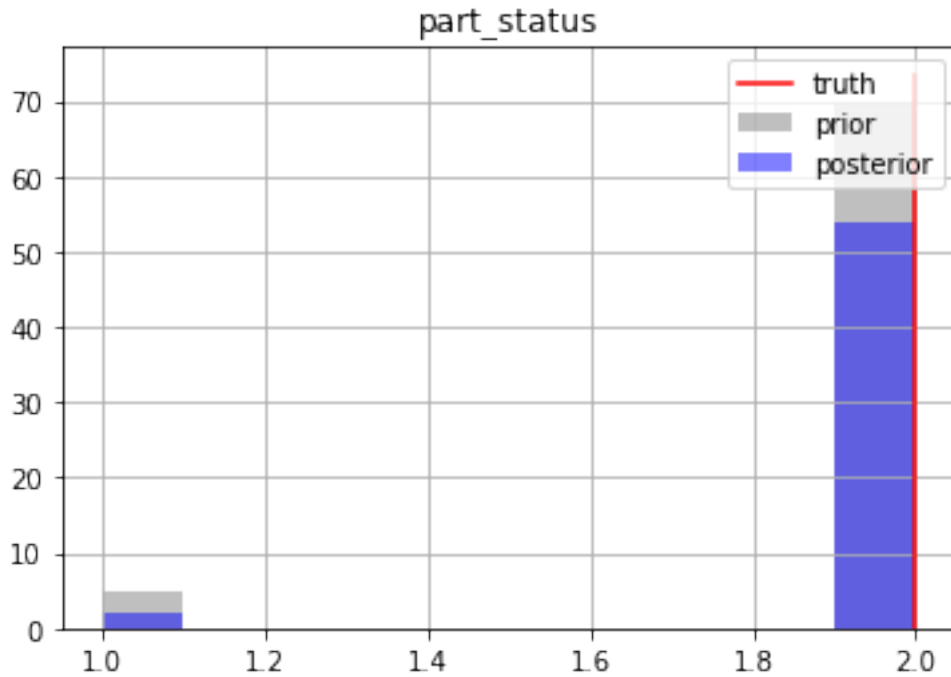
```











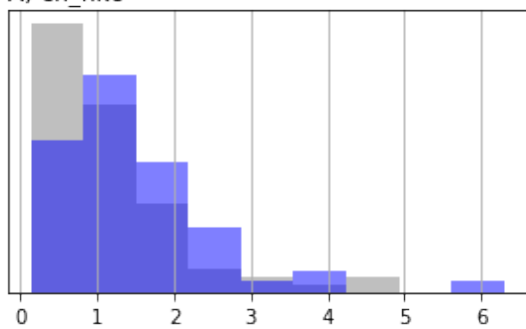
```
In [17]: pe_pr = pd.read_csv(os.path.join(m_d,"freyberg_ies.0.par.csv"),index_col=0)
pe_pt = pd.read_csv(os.path.join(m_d,"freyberg_ies.{0}.par.csv".format(pst.control_da
#pe_pr.index = pe_pt.index
#par = pst.parameter_data
print(pe_pr.shape,pe_pt.shape)
pdict = par.groupby("pargp").groups
pyemu.plot_utils.ensemble_helper({"0.5":pe_pr,"b":pe_pt},plot_cols=pdict)
#pyemu.plot_utils.ensemble_change_summary(pe_pr,pe_pt,pst=pst,bins=20)
```

```
/Users/jeremyw/miniconda3/lib/python3.5/site-packages/IPython/core/interactiveshell.py:2785: D
interactivity=interactivity, compiler=compiler, result=result)
```

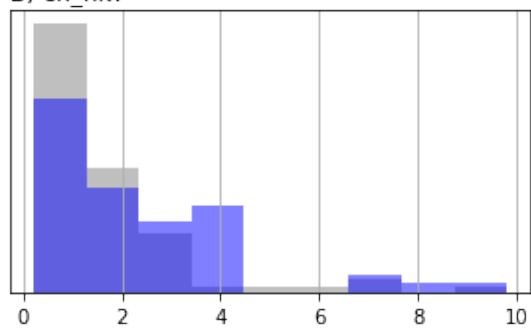
```
(75, 14819) (56, 14819)
```

```
<Figure size 576x756 with 0 Axes>
```

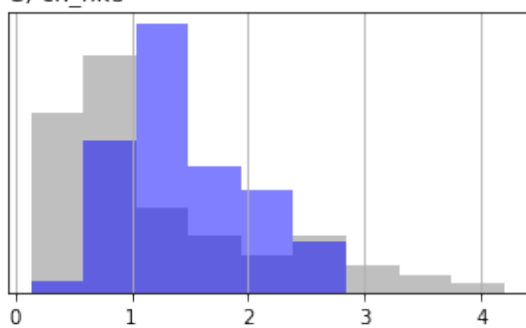
A) cn_hk6



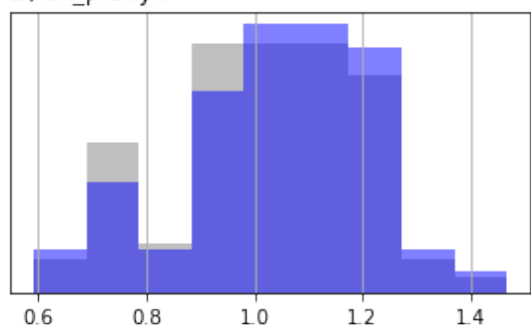
B) cn_hk7



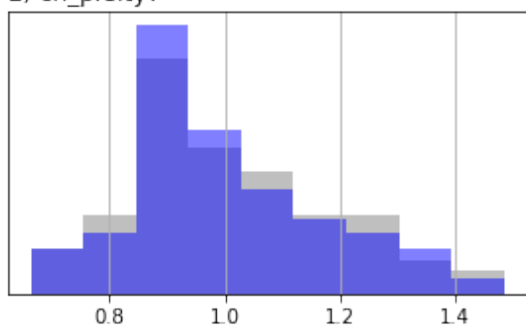
C) cn_hk8



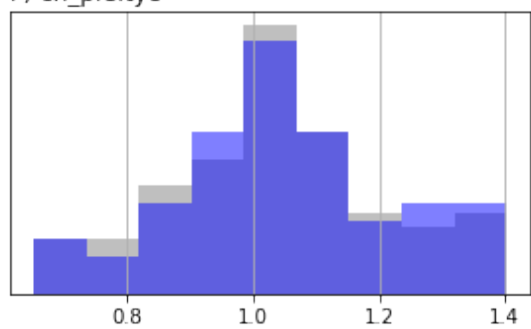
D) cn_prsity6



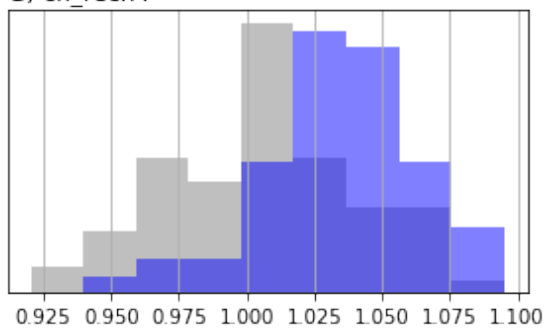
E) cn_prsity7



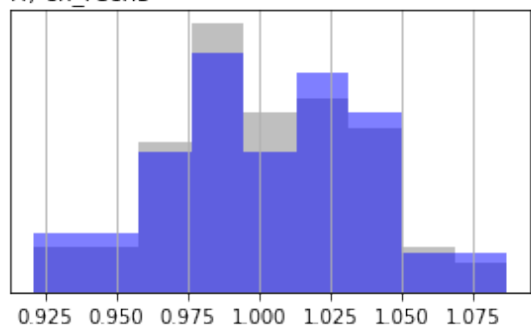
F) cn_prsity8

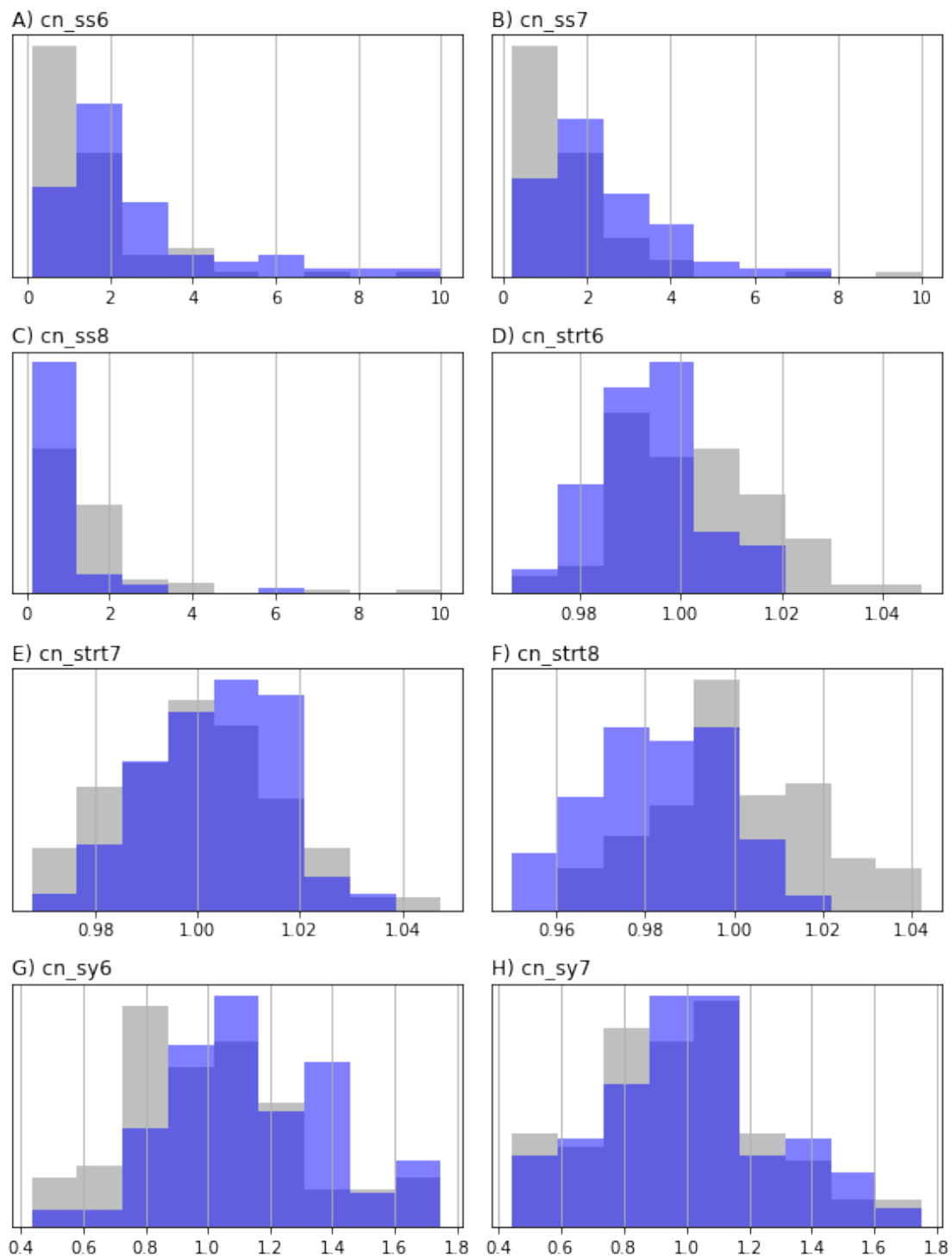


G) cn_rech4

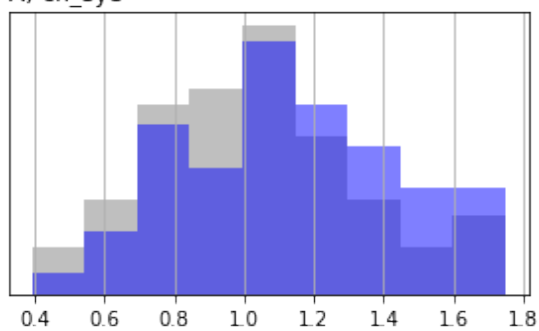


H) cn_rech5

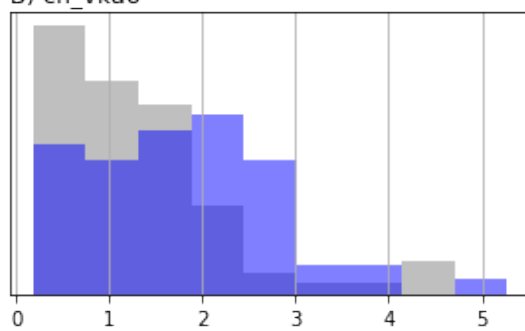




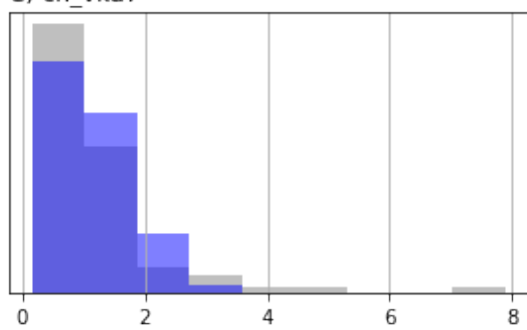
A) cn_sy8



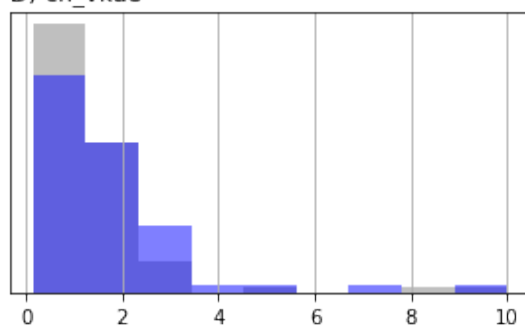
B) cn_vka6



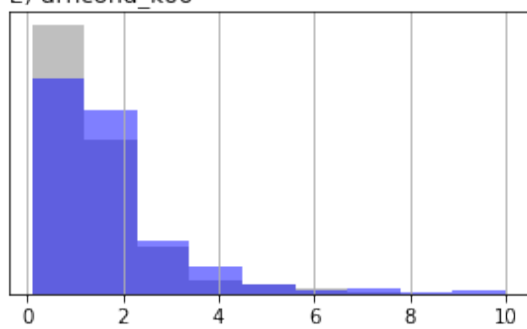
C) cn_vka7



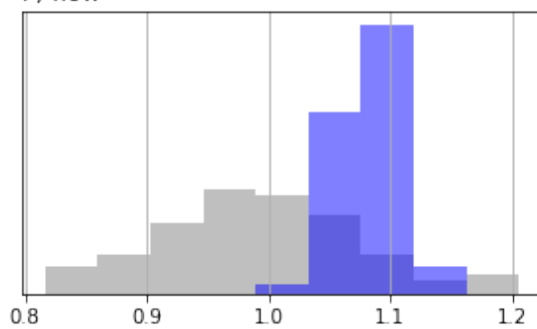
D) cn_vka8



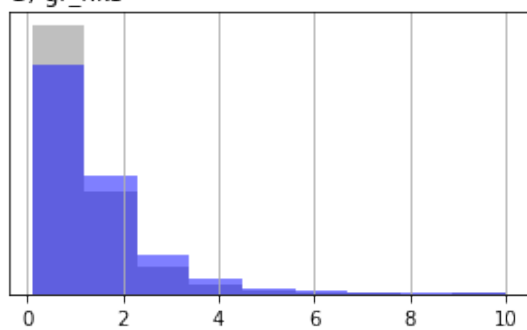
E) drncond_k00



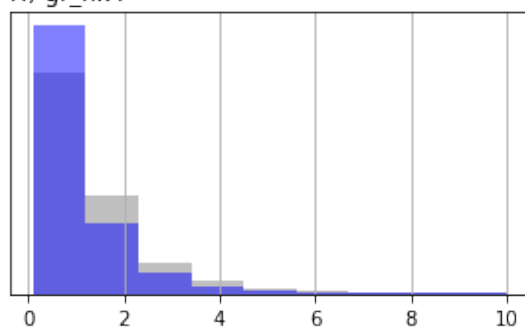
F) flow



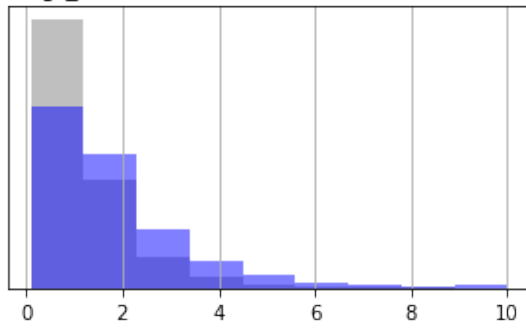
G) gr_hk3



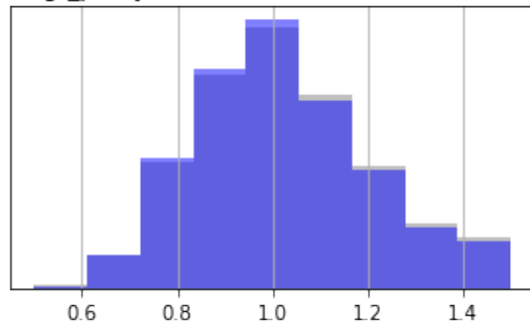
H) gr_hk4



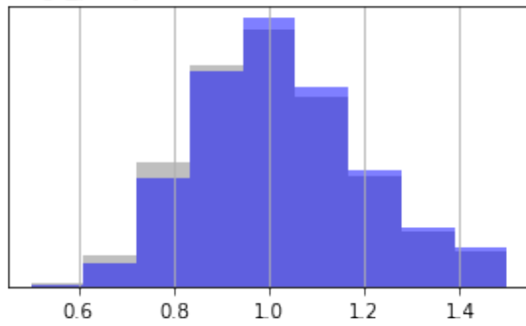
A) gr_hk5



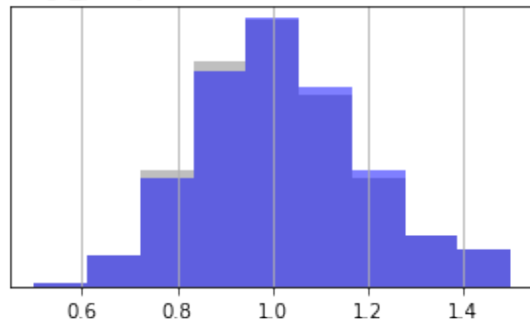
B) gr_prsity3



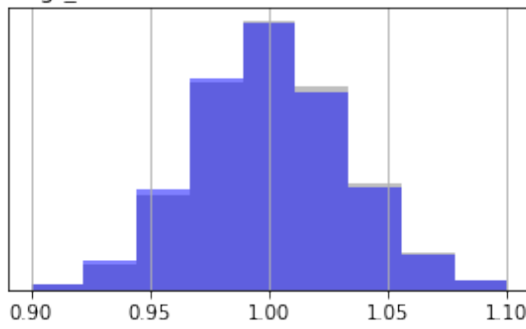
C) gr_prsity4



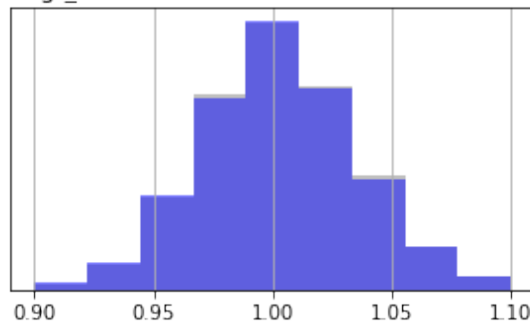
D) gr_prsity5



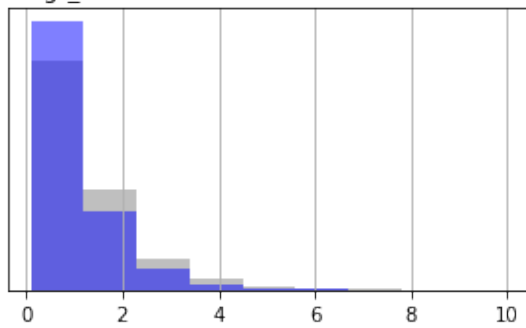
E) gr_rech2



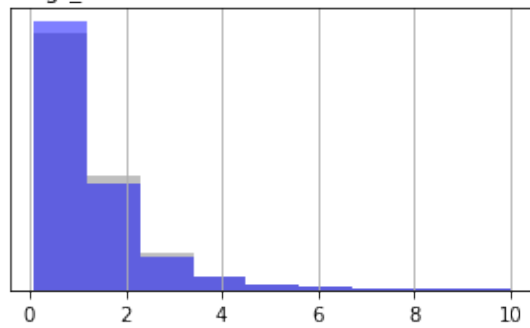
F) gr_rech3



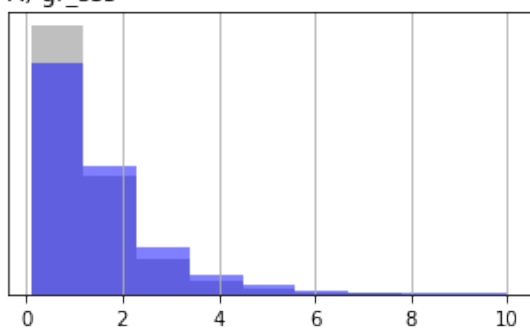
G) gr_ss3



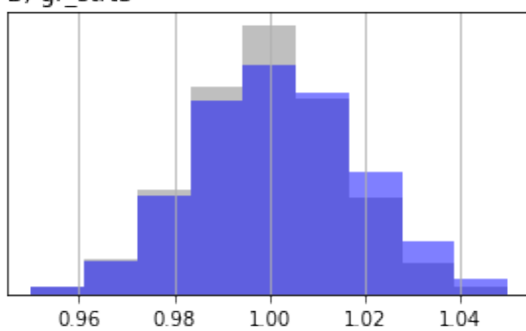
H) gr_ss4



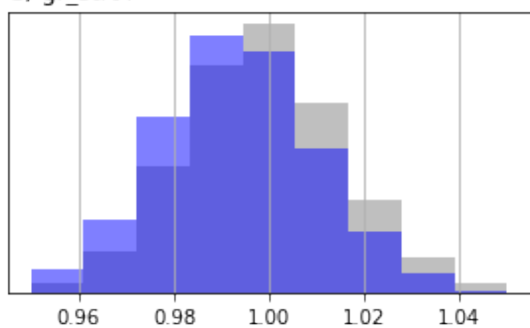
A) gr_ss5



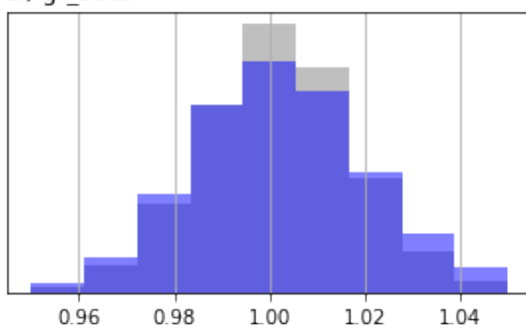
B) gr_strt3



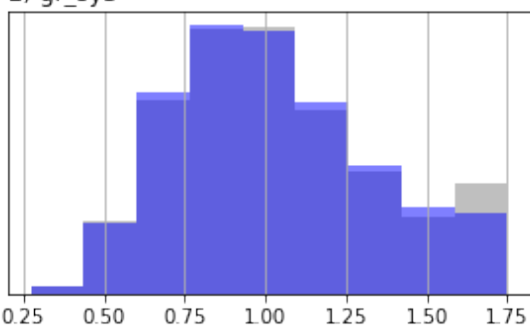
C) gr_strt4



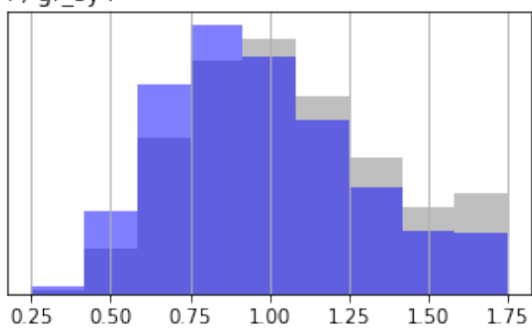
D) gr_strt5



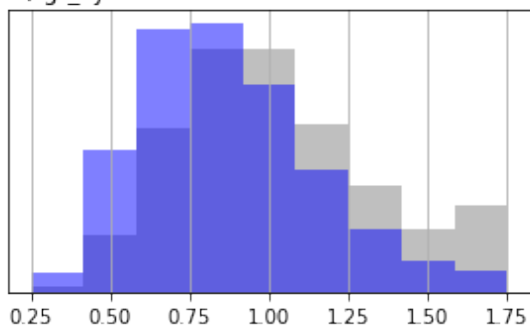
E) gr_sy3



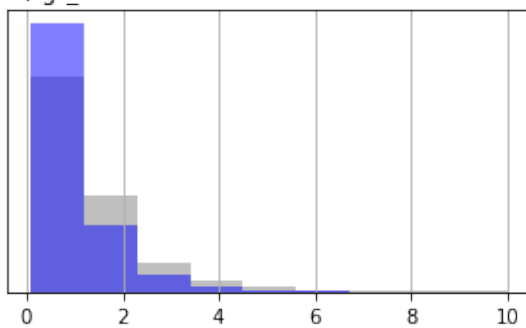
F) gr_sy4



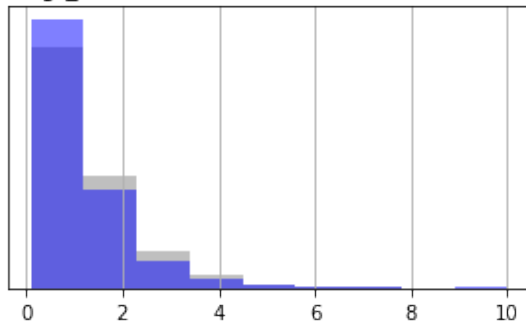
G) gr_sy5



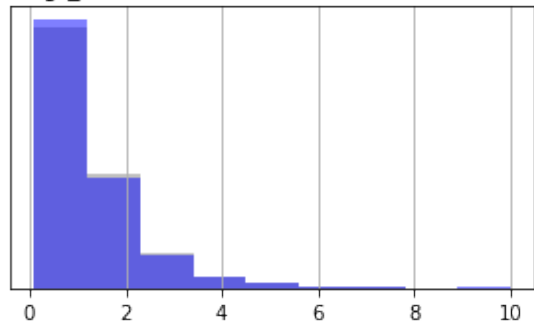
H) gr_vka3



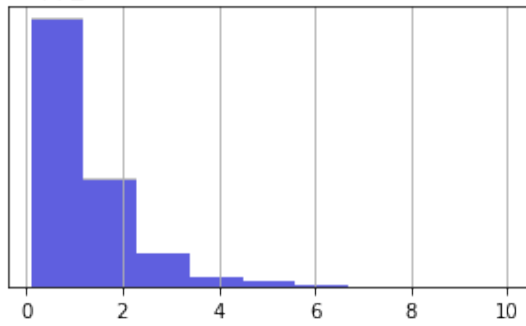
A) gr_vka4



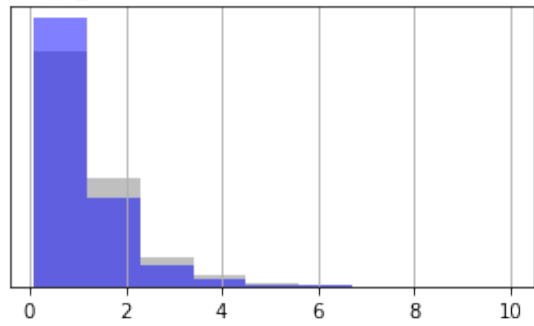
B) gr_vka5



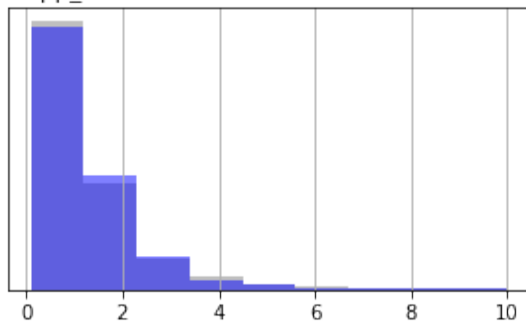
C) pp_hk0



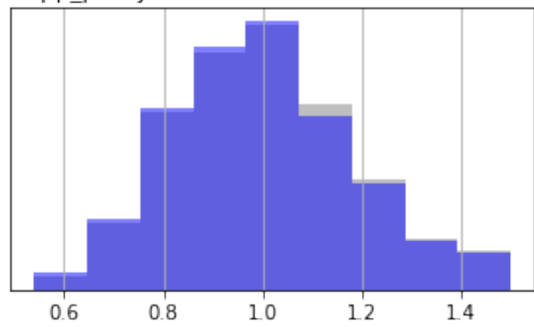
D) pp_hk1



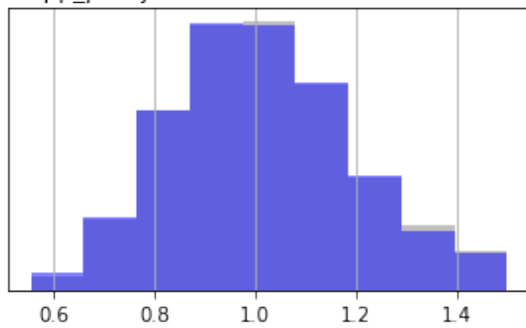
E) pp_hk2



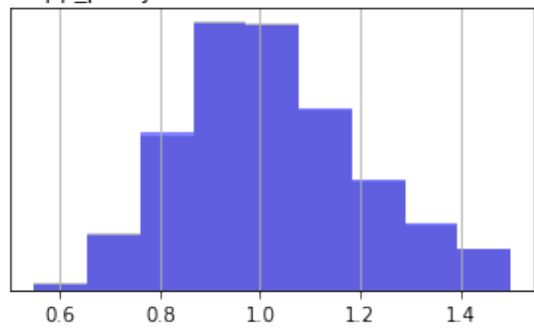
F) pp_prsity0



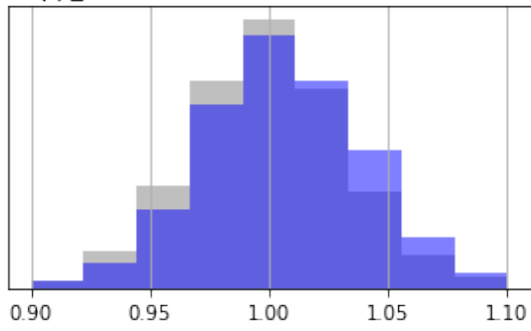
G) pp_prsity1



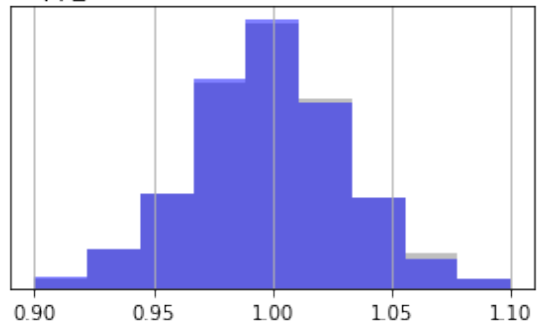
H) pp_prsity2



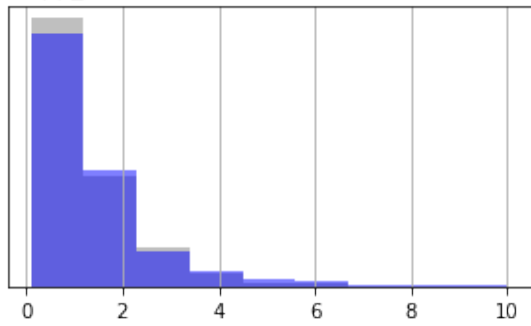
A) pp_rech0



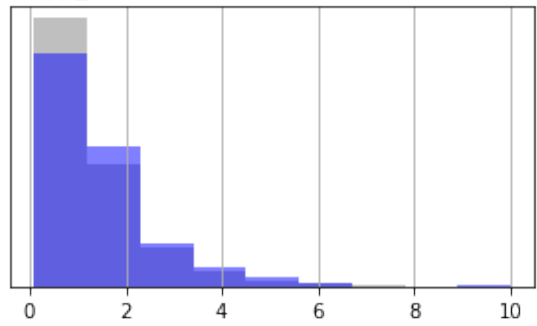
B) pp_rech1



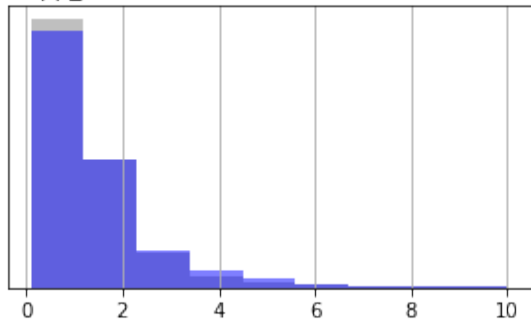
C) pp_ss0



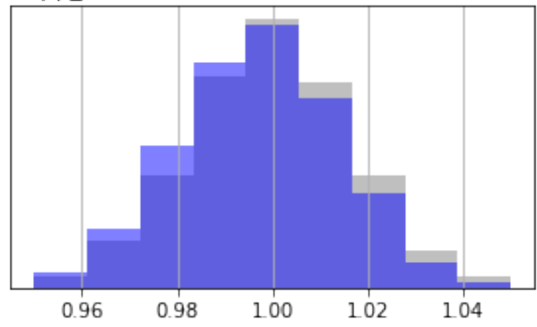
D) pp_ss1



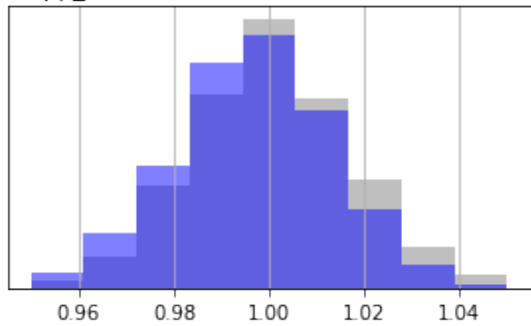
E) pp_ss2



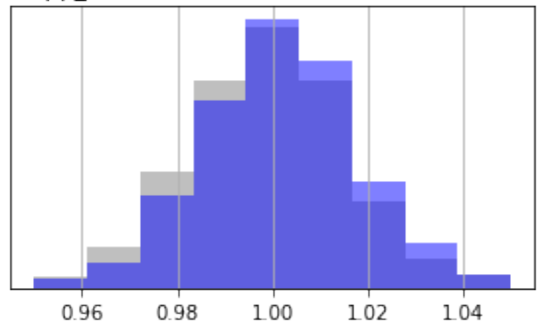
F) pp_strt0



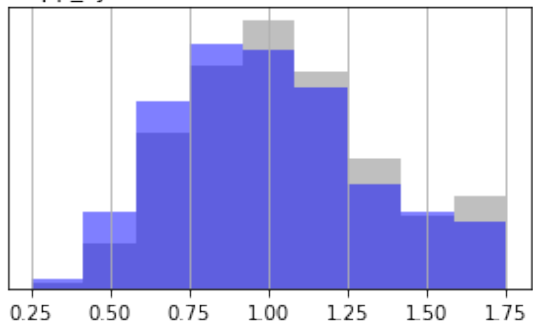
G) pp_strt1



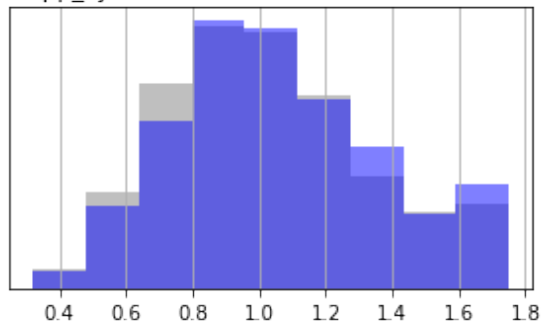
H) pp_strt2



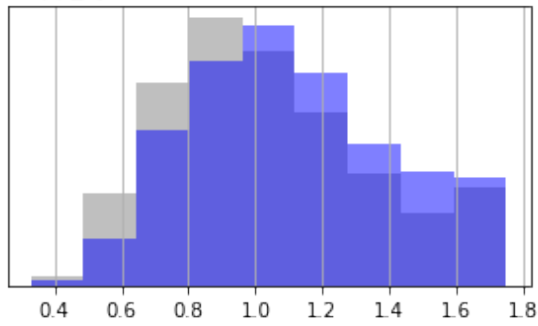
A) pp_sy0



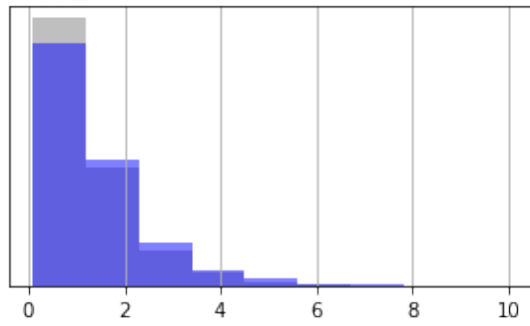
B) pp_sy1



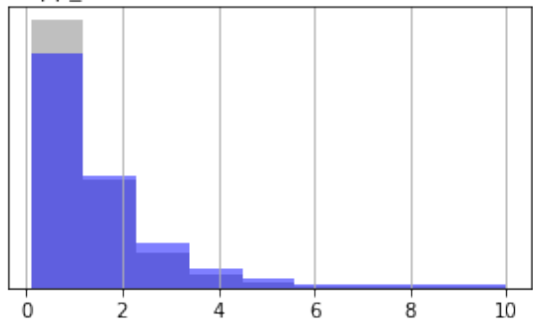
C) pp_sy2



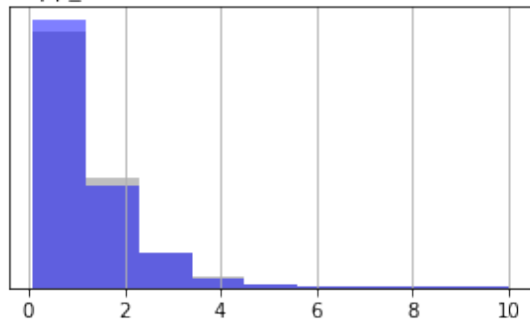
D) pp_vka0



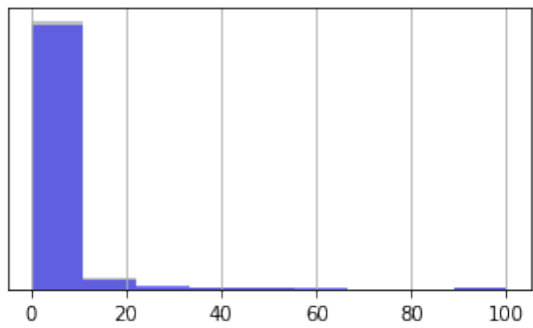
E) pp_vka1



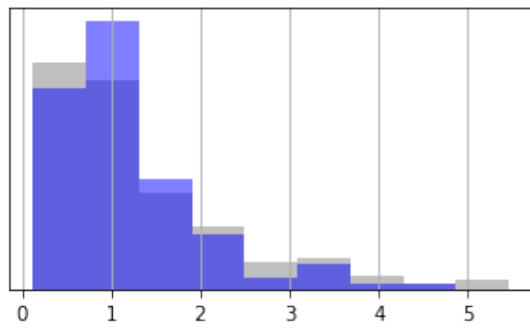
F) pp_vka2

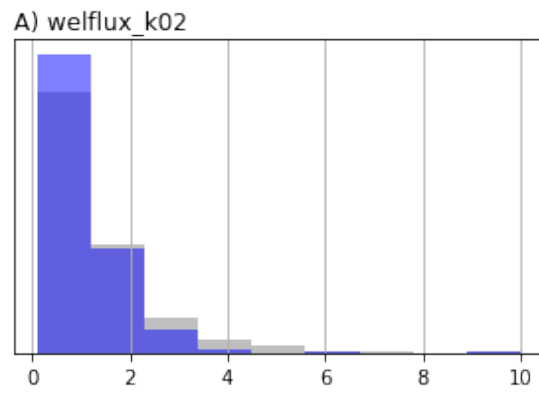


G) strk



H) welflux





1.1.3 PESTPP-IES with par-by-par distance based localization

```
In [18]: m = flopy.modflow.Modflow.load("freyberg.nam", model_ws="template")
```

```
In [19]: par = pst.parameter_data
```

```
gr_par = par.loc[par.pargp.apply(lambda x: "gr" in x and "prcity" not in x),:].copy()
print(gr_par.pargp.unique())
gr_par.groupby("pargp").groups
gr_par.loc[:, "i"] = gr_par.parnme.apply(lambda x: int(x[-6:-3]))
gr_par.loc[:, "j"] = gr_par.parnme.apply(lambda x: int(x[-3:]))
gr_par.loc[:, "x"] = gr_par.apply(lambda x: m.sr.xcentergrid[x.i,x.j],axis=1)
gr_par.loc[:, "y"] = gr_par.apply(lambda x: m.sr.ycentergrid[x.i,x.j],axis=1)
```

```
obs = pst.observation_data
```

```
nobs = obs.loc[obs.obgnme=="calhead",:].copy()
nobs.loc[:, "i"] = nobs.obsnme.apply(lambda x: int(x.split('_')[2]))
nobs.loc[:, "j"] = nobs.obsnme.apply(lambda x: int(x.split('_')[3]))
nobs.loc[:, "x"] = nobs.apply(lambda x: m.sr.xcentergrid[x.i,x.j],axis=1)
nobs.loc[:, "y"] = nobs.apply(lambda x: m.sr.ycentergrid[x.i,x.j],axis=1)
```

```
pp_tpl = [f for f in os.listdir(t_d) if "pp" in f and f.endswith(".tpl")]
pp_tpl_dfs = [pyemu.pp_utils.pp_tpl_to_dataframe(os.path.join(t_d,f)) for f in pp_tpl]
pp_par = pd.concat(pp_tpl_dfs)
pp_par.index = pp_par.parnme
#pp_par = par.loc[par.pargp.apply(lambda x: "pp" in x),:].copy()
```

```
['gr_hk3' 'gr_hk4' 'gr_hk5' 'gr_rech2' 'gr_rech3' 'gr_ss3' 'gr_ss4'
'gr_ss5' 'gr_strt3' 'gr_strt4' 'gr_strt5' 'gr_sy3' 'gr_sy4' 'gr_sy5'
'gr_vka3' 'gr_vka4' 'gr_vka5']
```

1.2 We will set up localization such that parameters are only informed by observations within a user-specified distance (we can use 5000 meters)

```
In [20]: loc = pyemu.Matrix.from_names(pst.nnz_obs_names,pst.adj_par_names).to_dataframe()
loc.loc[:, :] = 1.0
loc_dist = 5000.0
sadj = set(pst.adj_par_names)
print('obsname          fraction grid retained   fraction pilot points retained')
for oname in obs.loc[obs.obgnme=="calhead", "obsnme"]:
    xx,yy = nobs.loc[oname, ['x', 'y']]

    # localization for grid-based parameters
    gr_par.loc[:, "dist"] = gr_par.apply(lambda x: (x.x - xx)**2 + (x.y - yy)**2,axis=1)
    gr_too_far = gr_par.loc[gr_par.dist > loc_dist, "parname"]
    gr_too_far = gr_too_far.loc[gr_too_far.apply(lambda x: x in sadj)]
    loc.loc[:, gr_too_far] = 0.0

    # localization for pilot point parameters
    pp_par.loc[:, "dist"] = pp_par.apply(lambda x: (x.x - xx)**2 + (x.y - yy)**2,axis=1)
```

```

pp_too_far = pp_par.loc[pp_par.dist > loc_dist,"parname"]
pp_too_far = pp_too_far.loc[pp_too_far.apply(lambda x: x in sadj)]
loc.loc[oname,pp_too_far] = 0.0
print(oname,gr_too_far.shape[0]/gr_par.shape[0],pp_too_far.shape[0]/pp_par.shape[0])

loc.loc[:,dont_pars] = 0.0
#spars = par.loc[par.parnme.apply(lambda x: "ss" in x or "sy" in x),"parname"]
#loc.loc[:,spars] = 0.0
print('\n\nTotal number of parameters still informed by each observation')
loc.sum(axis=1)

```

obsname	fraction grid retained	fraction pilot points retained
hds_00_002_009_000	0.46382978723404256	0.46875
hds_00_002_015_000	0.4794326241134752	0.5
hds_00_003_008_000	0.43829787234042555	0.34375
hds_00_009_001_000	0.3304964539007092	0.25
hds_00_013_010_000	0.15319148936170213	0.09375
hds_00_015_016_000	0.13900709219858157	0.0625
hds_00_021_010_000	0.06950354609929078	0.03125
hds_00_022_015_000	0.12198581560283688	0.15625
hds_00_024_004_000	0.17872340425531916	0.15625
hds_00_026_006_000	0.2198581560283688	0.21875
hds_00_029_015_000	0.29929078014184396	0.28125
hds_00_033_007_000	0.3829787234042553	0.375
hds_00_034_010_000	0.4	0.40625

Total number of parameters still informed by each observation

```

Out[20]: fo_39_19791230      1786.0
         hds_00_002_009_000    1546.0
         hds_00_002_015_000    1530.0
         hds_00_003_008_000    1610.0
         hds_00_009_001_000    1658.0
         hds_00_013_010_000    1738.0
         hds_00_015_016_000    1754.0
         hds_00_021_010_000    1770.0
         hds_00_022_015_000    1706.0
         hds_00_024_004_000    1706.0
         hds_00_026_006_000    1674.0
         hds_00_029_015_000    1642.0
         hds_00_033_007_000    1594.0
         hds_00_034_010_000    1578.0
         dtype: float64

```

```

In [21]: pyemu.Matrix.from_dataframe(loc).to_coo(os.path.join(t_d,"loc.jcb"))
         pst.pstpp_options["ies_localizer"] = "loc.jcb"
         pst.write(os.path.join(t_d,"freyberg_ies.pst"))

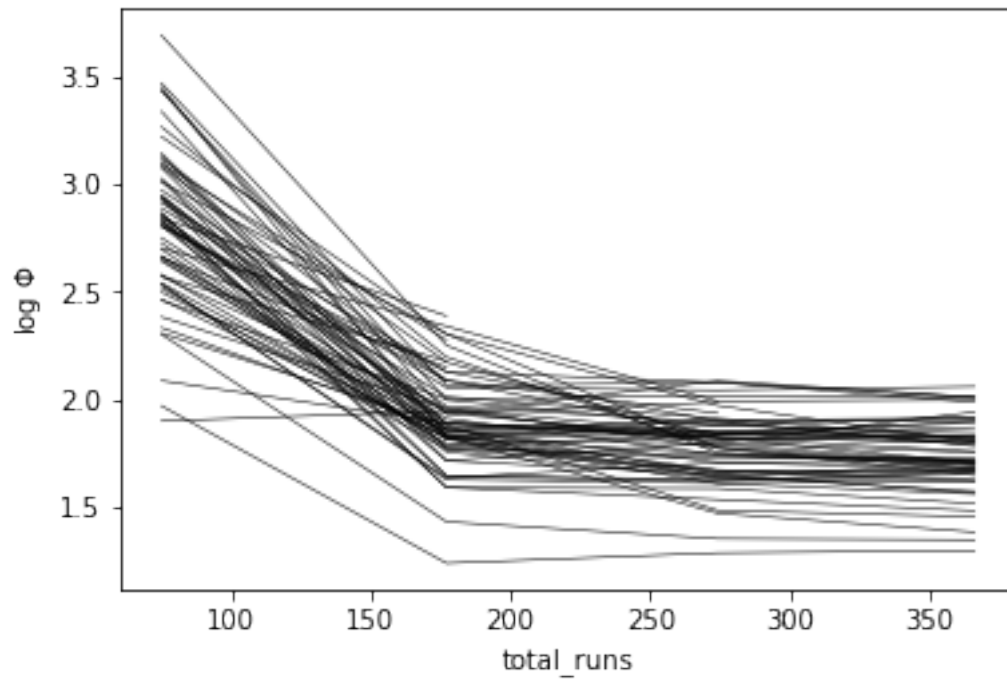
```

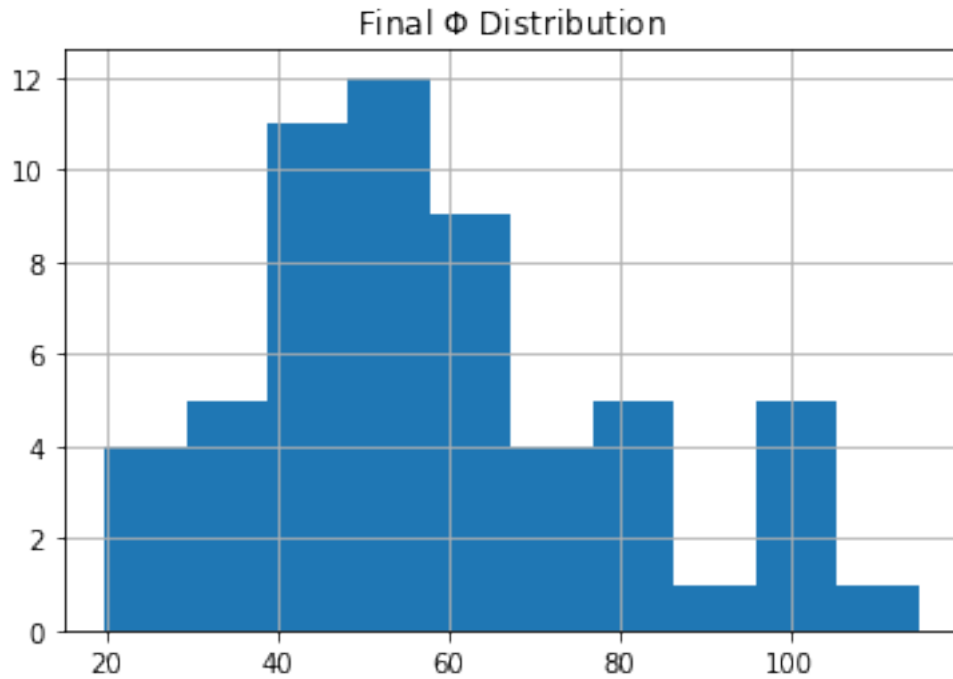
```
noptmax:3, npar_adj:14819, nnz_obs:14
```

1.2.1 now let's run it

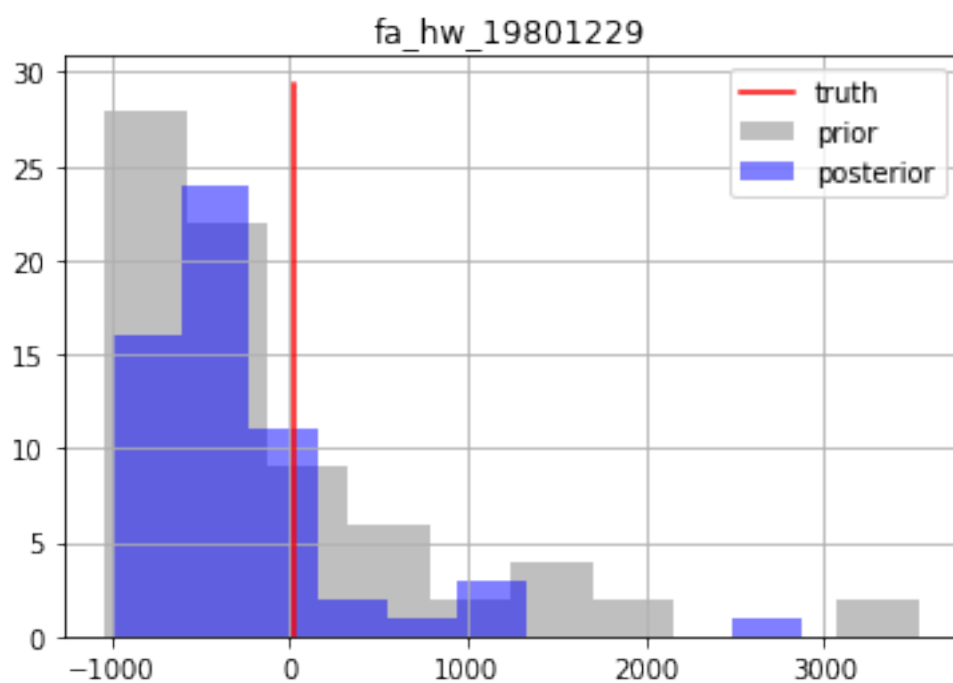
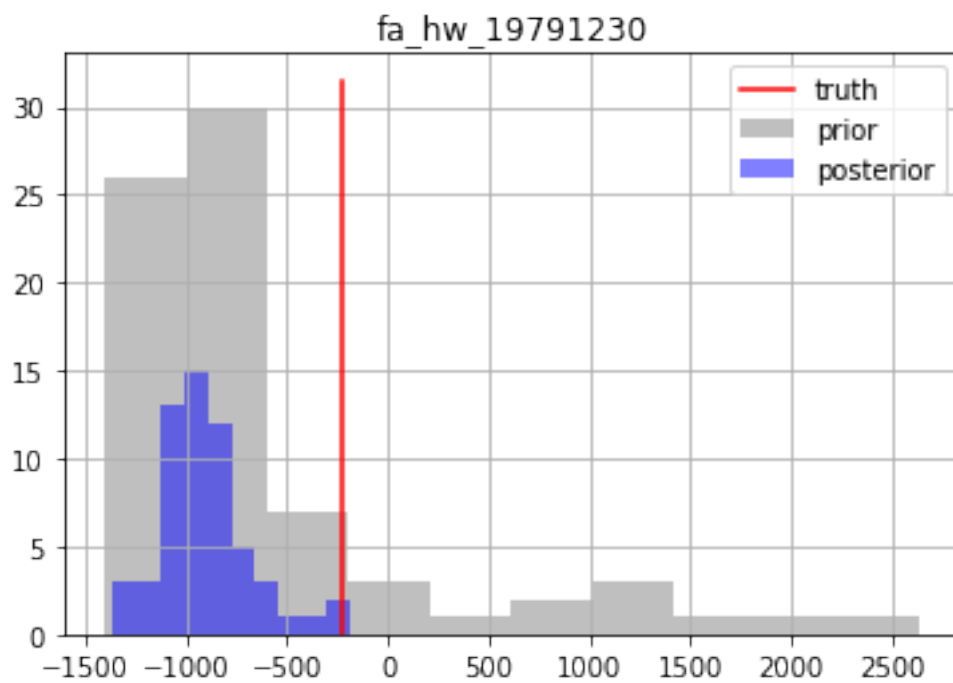
```
In [22]: pyemu.os_utils.start_slaves(t_d,"pestpp-ies","freyberg_ies.pst",num_slaves=num_workers)
```

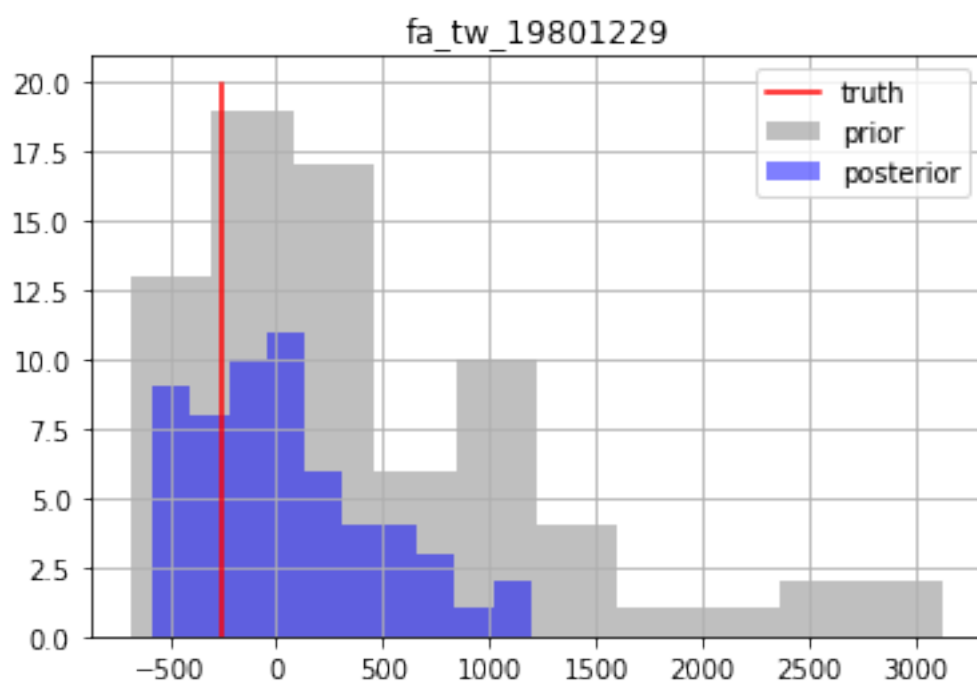
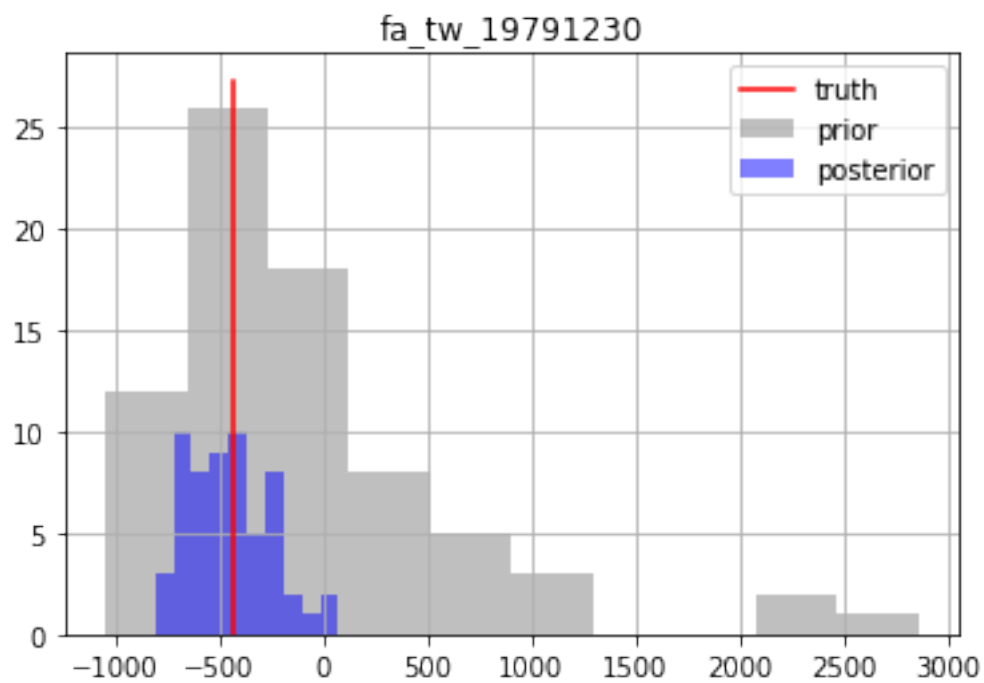
```
In [23]: phi = pd.read_csv(os.path.join(m_d,"freyberg_ies.phi.actual.csv"),index_col=0)
phi.index = phi.total_runs
phi.iloc[:,6:].apply(np.log10).plot(legend=False,lw=0.5,color='k')
plt.ylabel('log  $\Phi$ ')
plt.show()
phi.iloc[-1,6:].hist()
plt.title('Final  $\Phi$  Distribution');
```

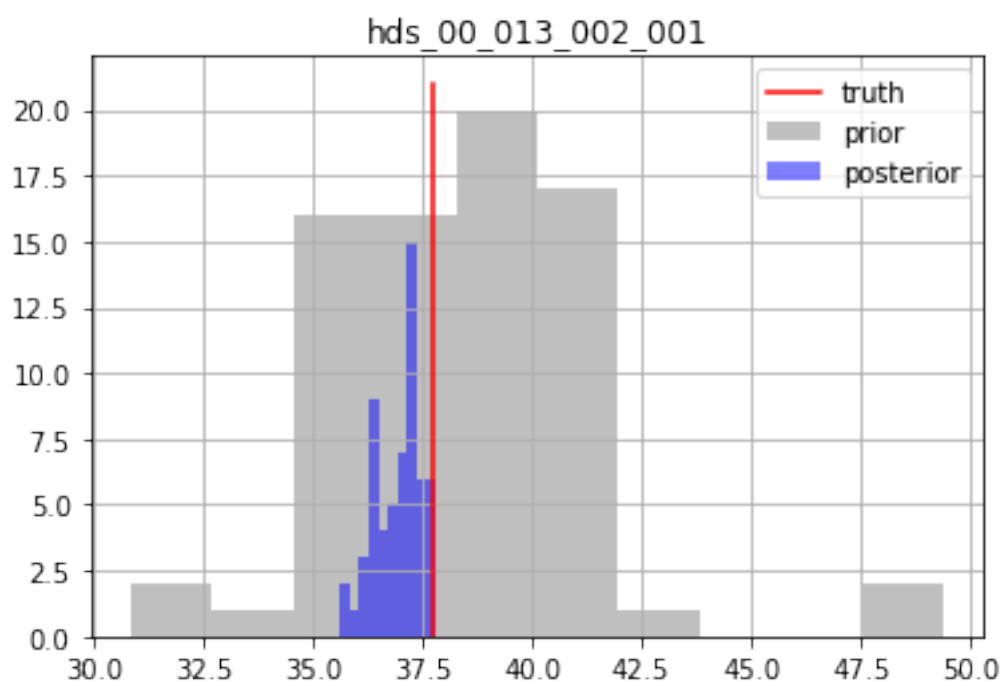
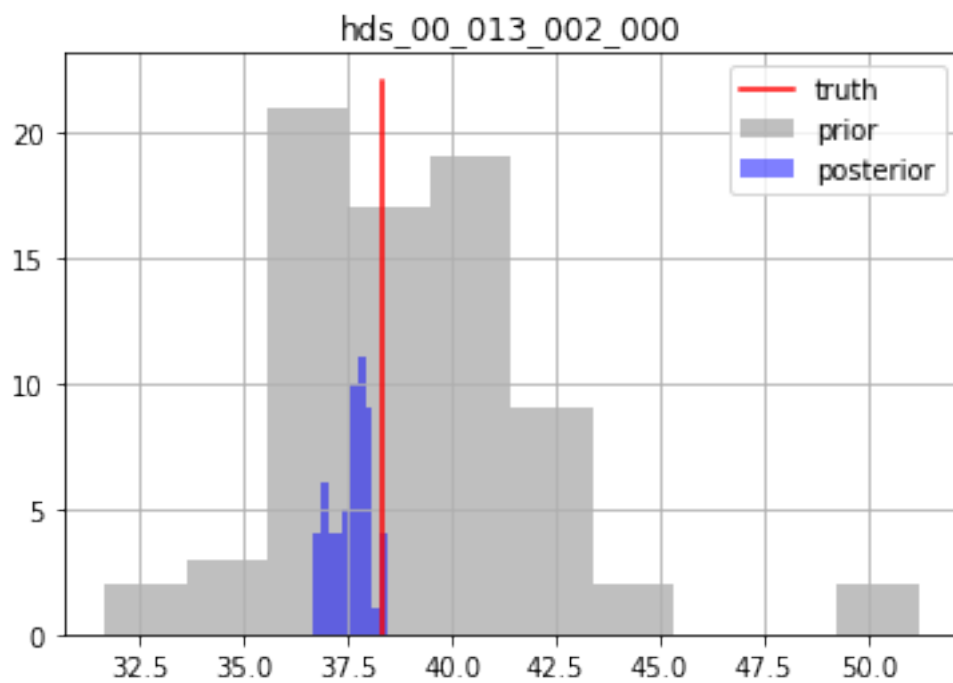


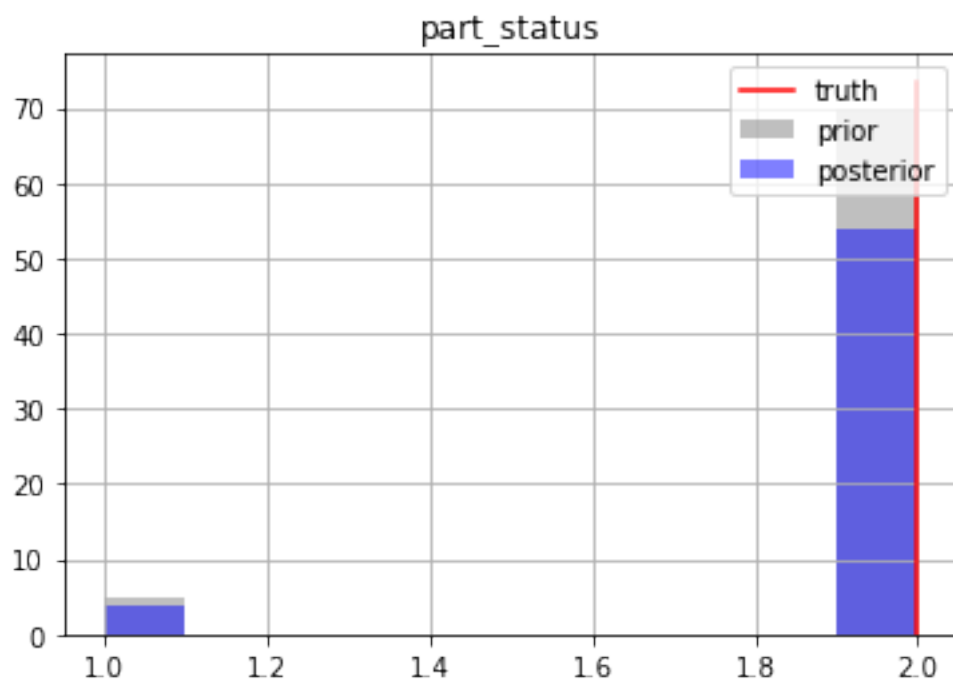
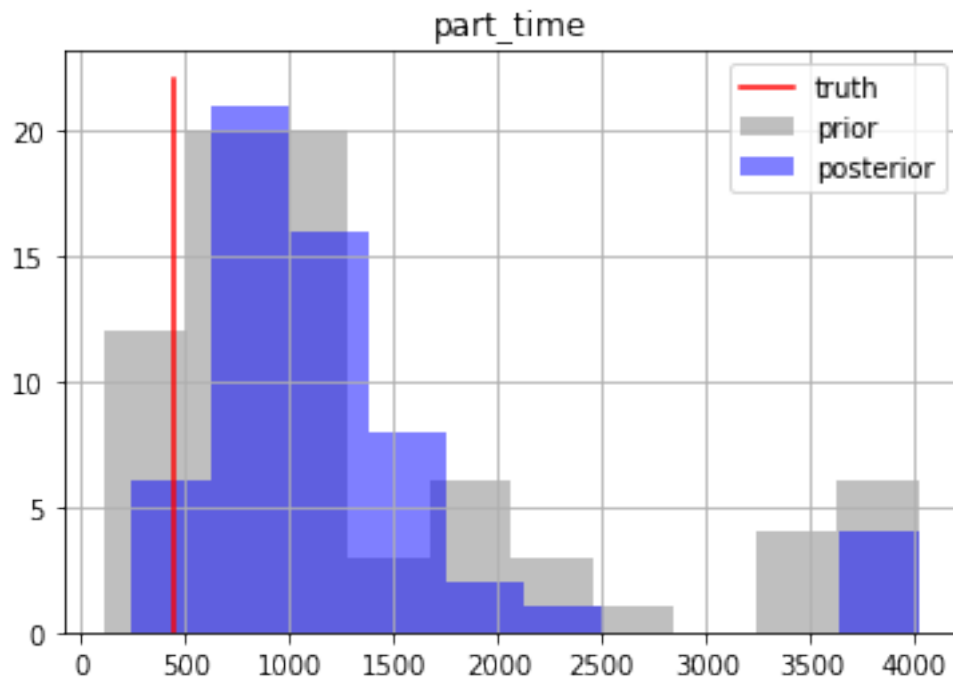


```
In [24]: oe_pr = pd.read_csv(os.path.join(m_d, "freyberg_ies.0.obs.csv"), index_col=0)
oe_pt = pd.read_csv(os.path.join(m_d, "freyberg_ies.{0}.obs.csv".format(pst.control_da
obs = pst.observation_data
fnames = pst.pestpp_options["forecasts"].split(",")
for forecast in fnames:
    ax = plt.subplot(111)
    oe_pr.loc[:, forecast].hist(ax=ax, color="0.5", alpha=0.5, label='prior')
    oe_pt.loc[:, forecast].hist(ax=ax, color="b", alpha=0.5, label='posterior')
    ax.plot([obs.loc[forecast, "obsval"], obs.loc[forecast, "obsval"]], ax.get_ylim(), "r")
    ax.set_title(forecast)
    ax.legend(loc='upper right')
    plt.show()
```







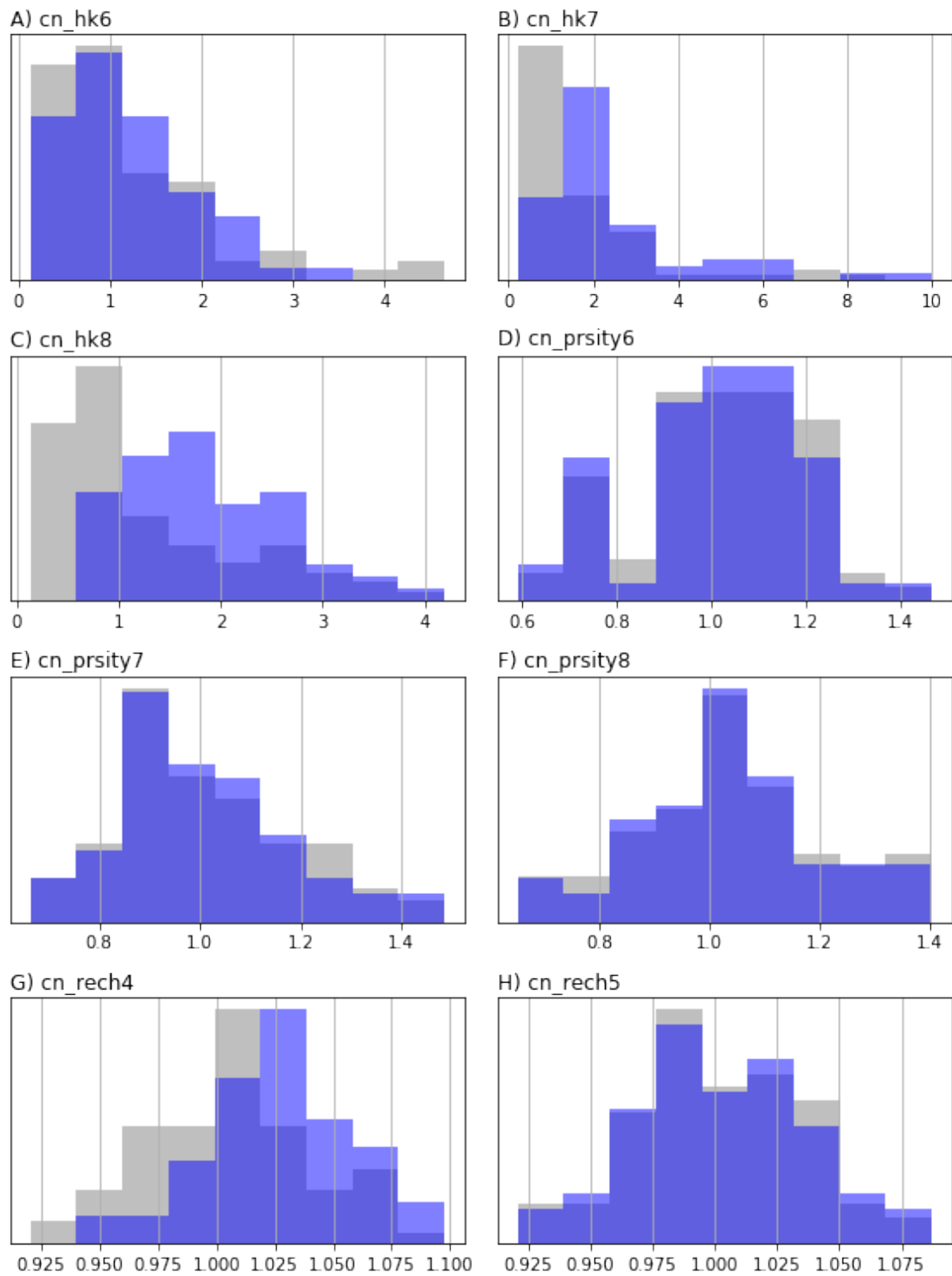


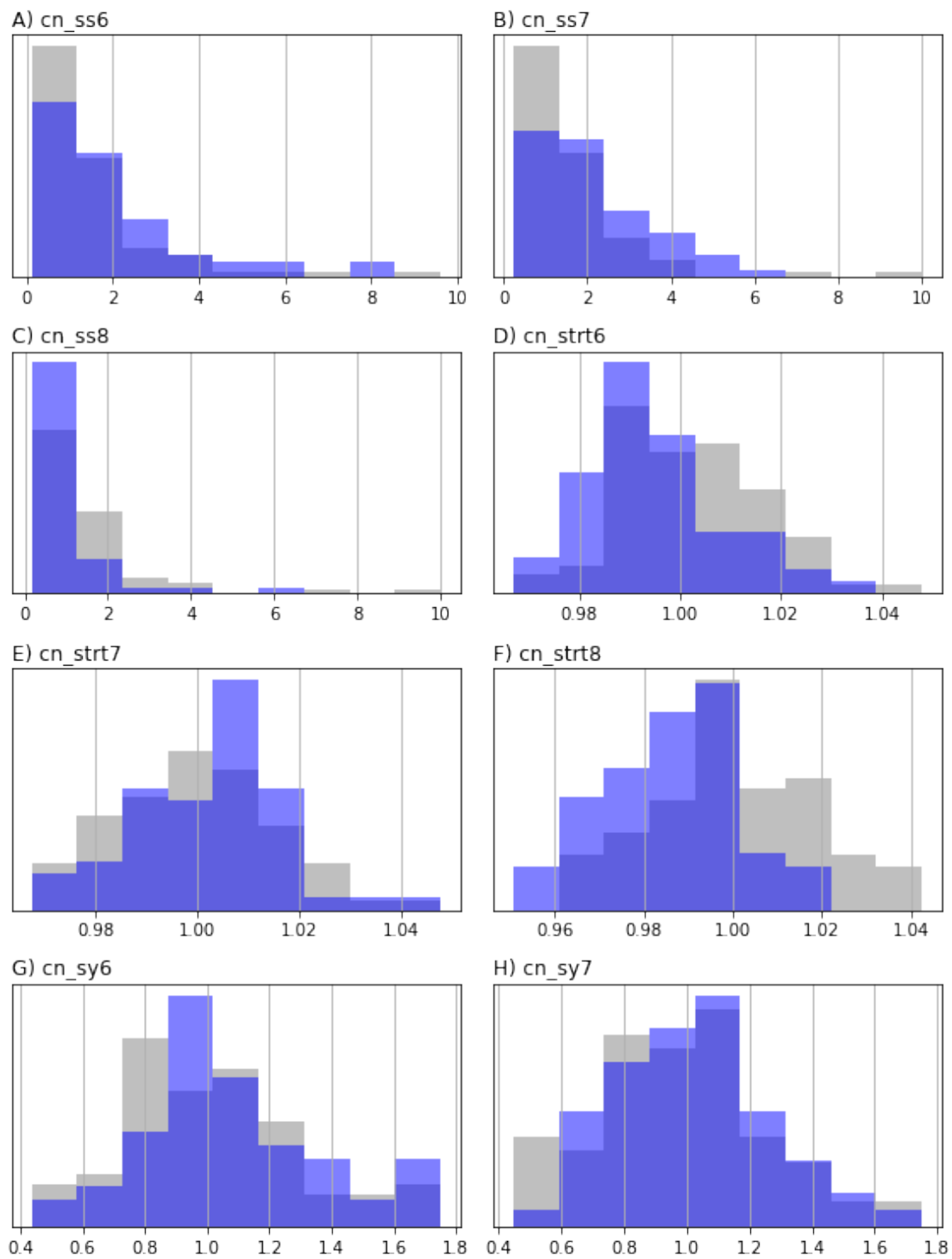
```
In [25]: pe_pr = pd.read_csv(os.path.join(m_d,"freyberg_ies.0.par.csv"),index_col=0)
         pe_pt = pd.read_csv(os.path.join(m_d,"freyberg_ies.{0}.par.csv".format(pst.control_da
```

```
par = pst.parameter_data
pdict = par.groupby("pargp").groups
pyemu.plot_utils.ensemble_helper({"0.5":pe_pr,"b":pe_pt},plot_cols=pdict)
#pyemu.plot_utils.ensemble_change_summary(pe_pr,pe_pt,pst=pst,bins=20)

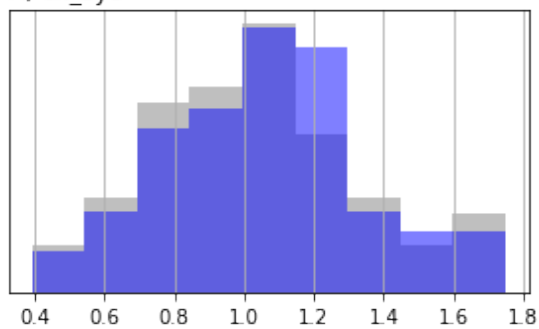
/Users/jeremyw/miniconda3/lib/python3.5/site-packages/IPython/core/interactiveshell.py:2785: D
interactivity=interactivity, compiler=compiler, result=result)
```

<Figure size 576x756 with 0 Axes>

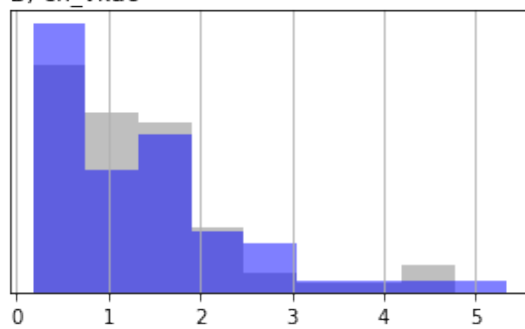




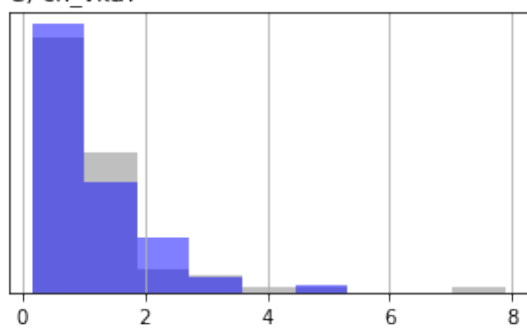
A) cn_sy8



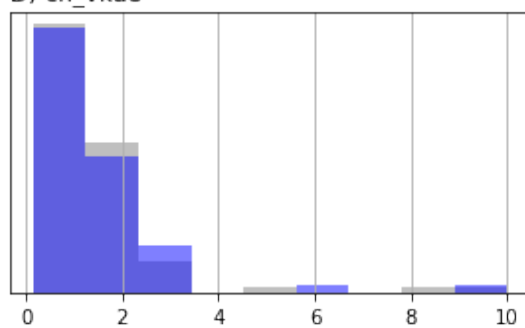
B) cn_vka6



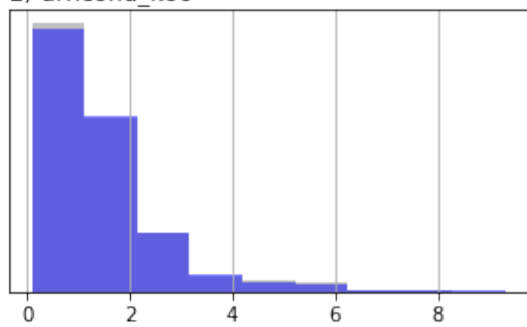
C) cn_vka7



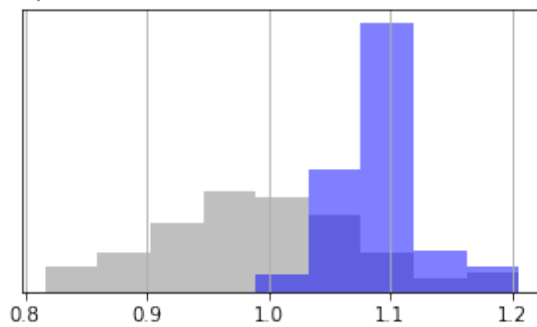
D) cn_vka8



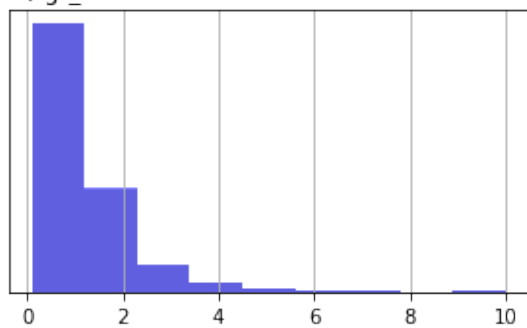
E) drncond_k00



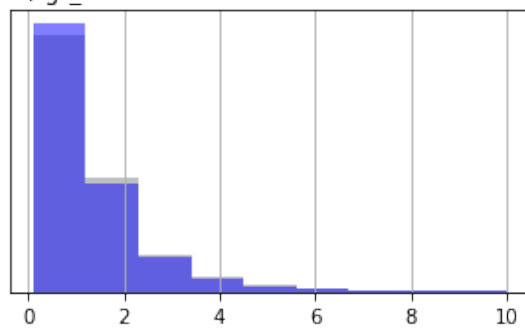
F) flow



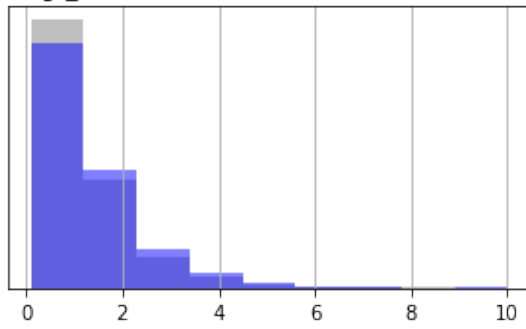
G) gr_hk3



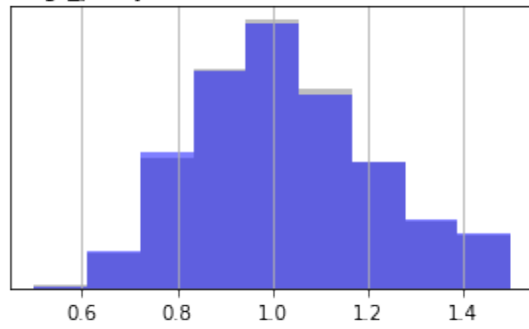
H) gr_hk4



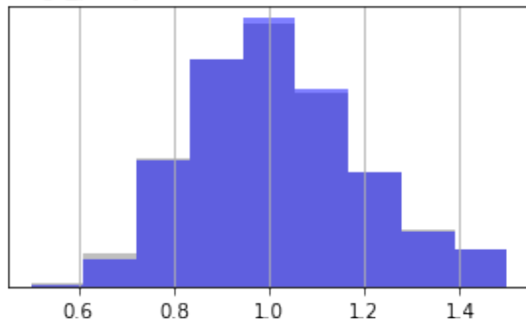
A) gr_hk5



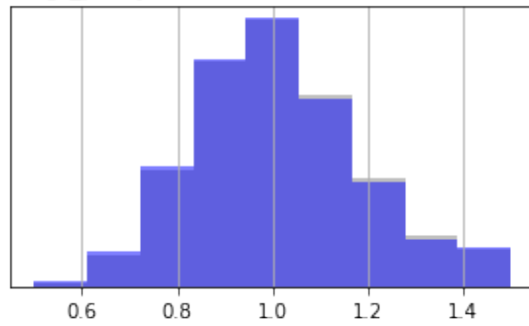
B) gr_prsity3



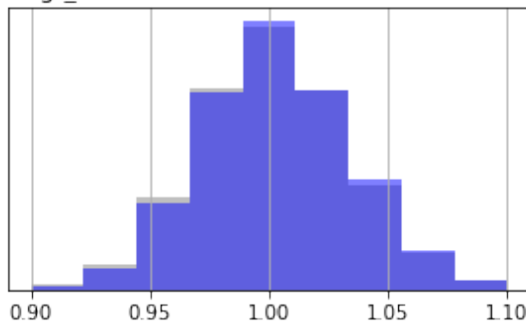
C) gr_prsity4



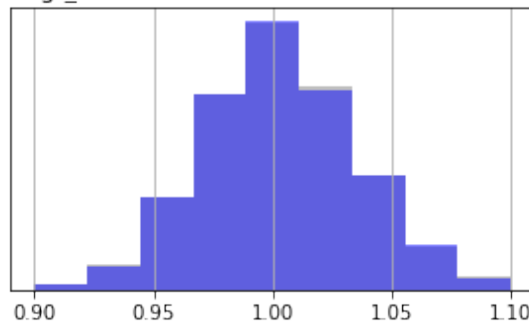
D) gr_prsity5



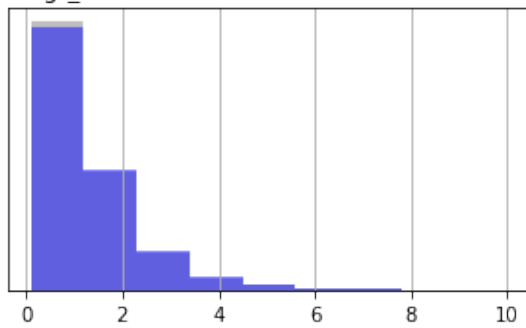
E) gr_rech2



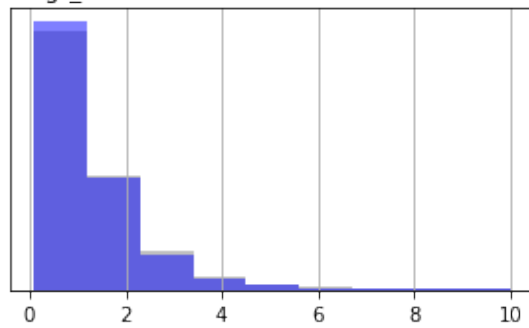
F) gr_rech3



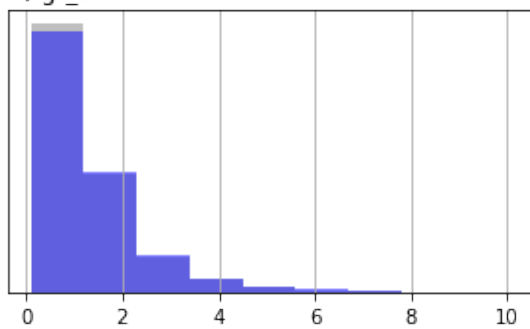
G) gr_ss3



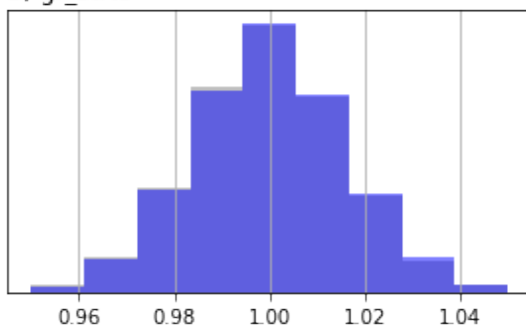
H) gr_ss4



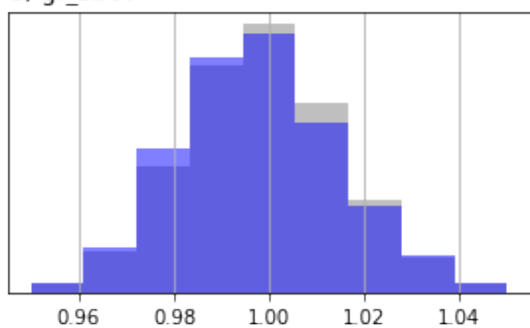
A) gr_ss5



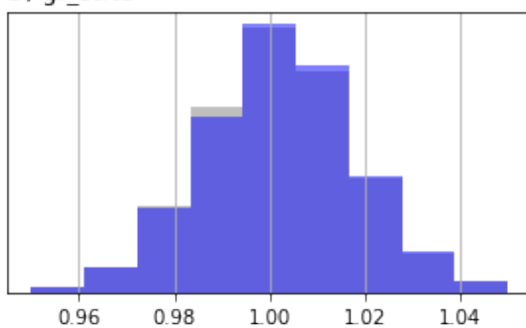
B) gr_strt3



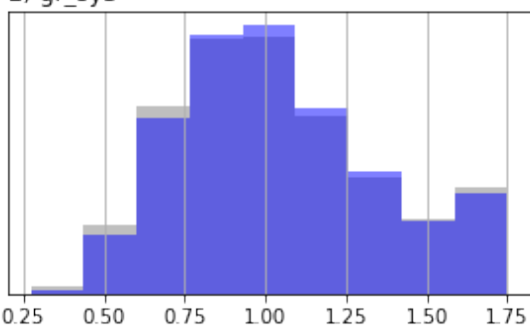
C) gr_strt4



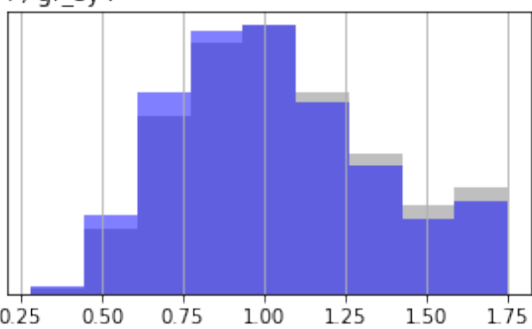
D) gr_strt5



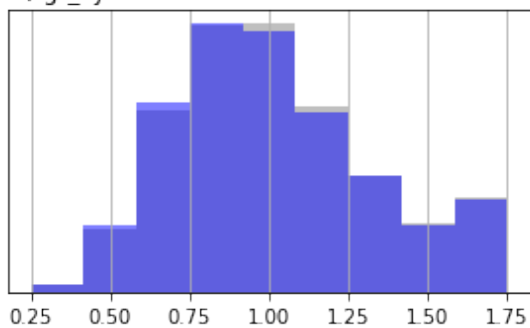
E) gr_sy3



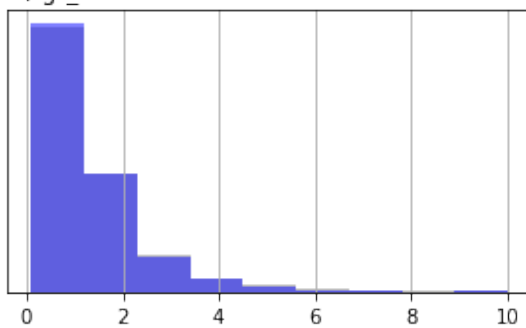
F) gr_sy4



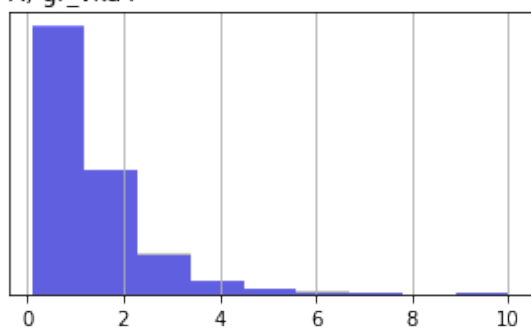
G) gr_sy5



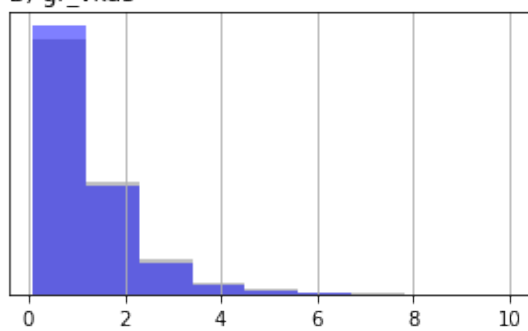
H) gr_vka3



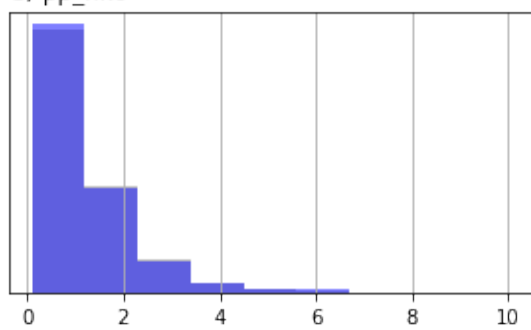
A) gr_vka4



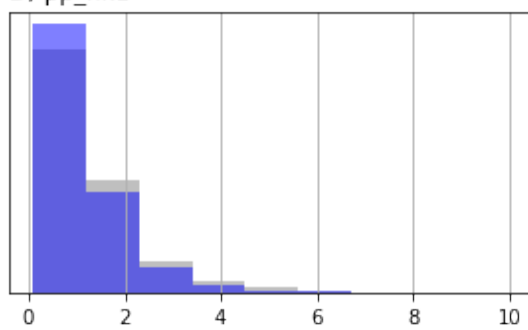
B) gr_vka5



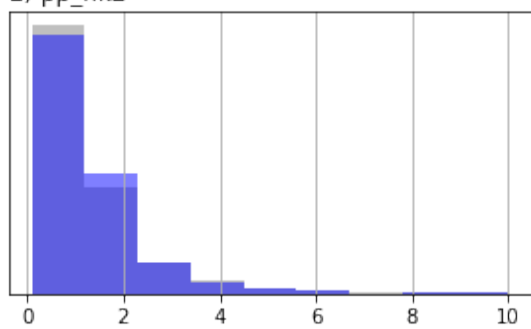
C) pp_hk0



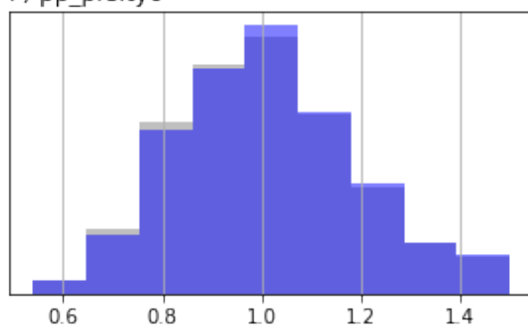
D) pp_hk1



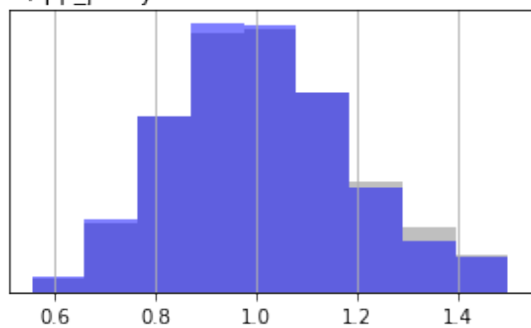
E) pp_hk2



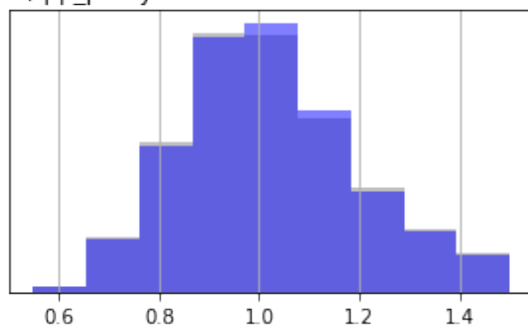
F) pp_prsity0



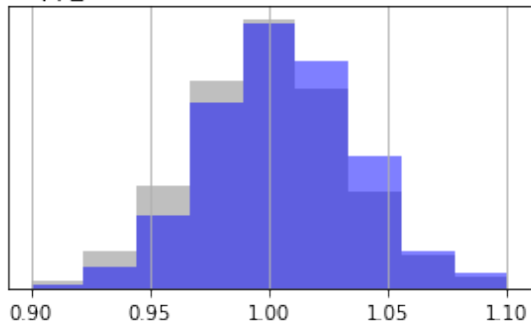
G) pp_prsity1



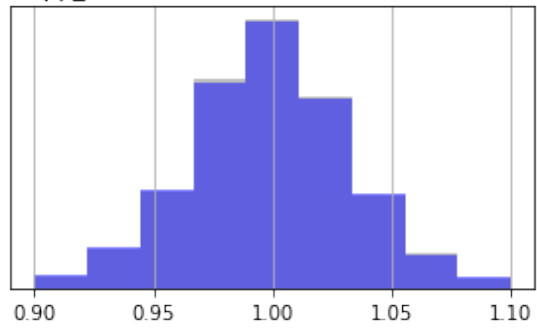
H) pp_prsity2



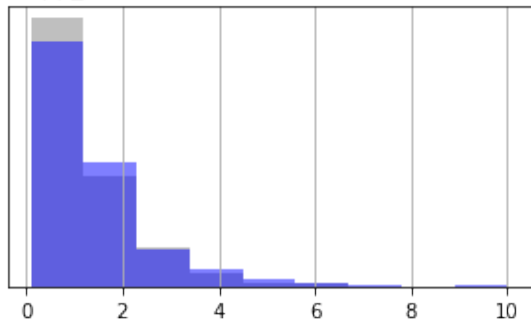
A) pp_rech0



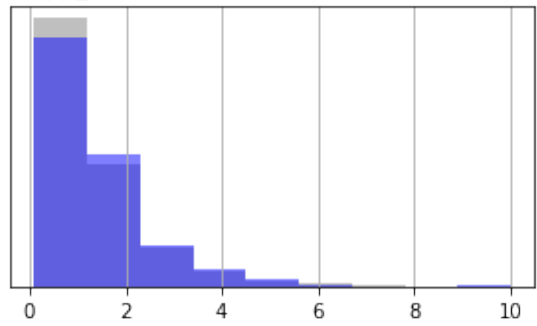
B) pp_rech1



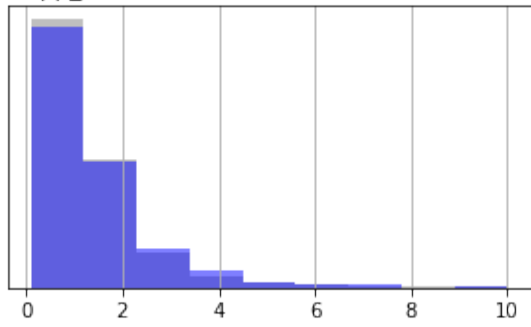
C) pp_ss0



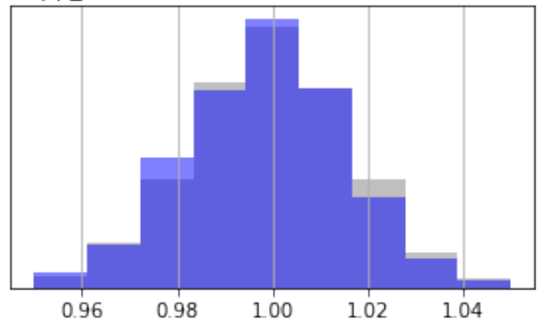
D) pp_ss1



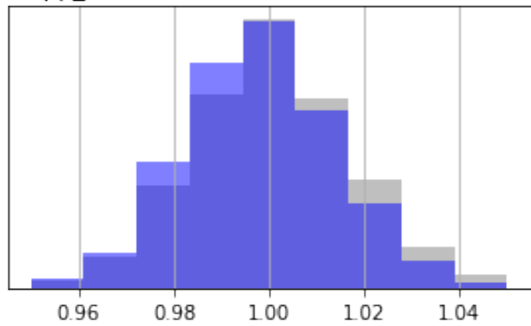
E) pp_ss2



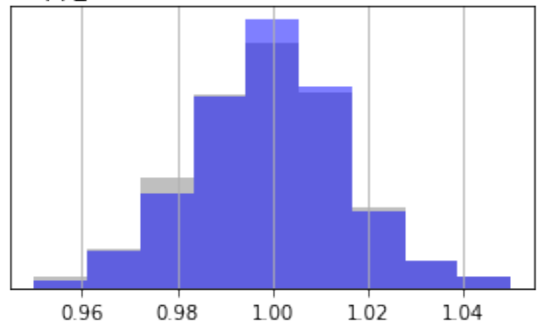
F) pp_strt0



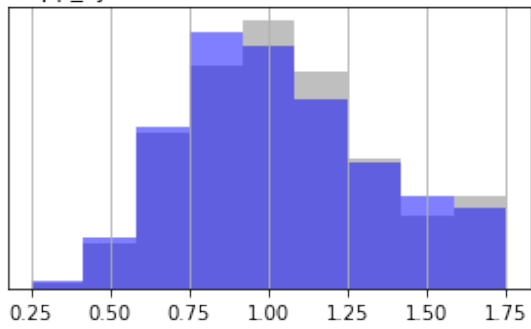
G) pp_strt1



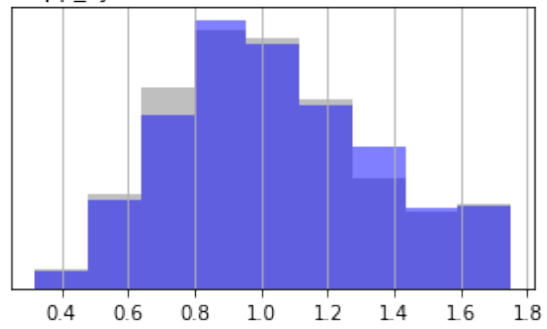
H) pp_strt2



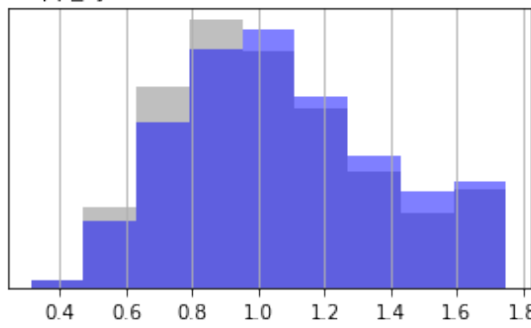
A) pp_sy0



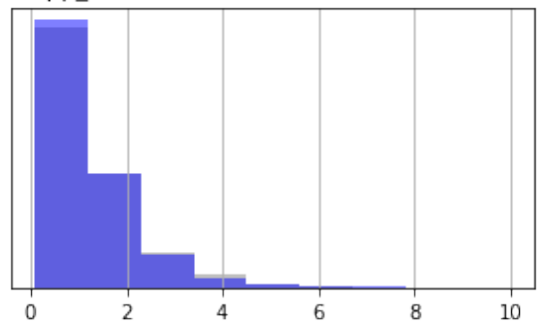
B) pp_sy1



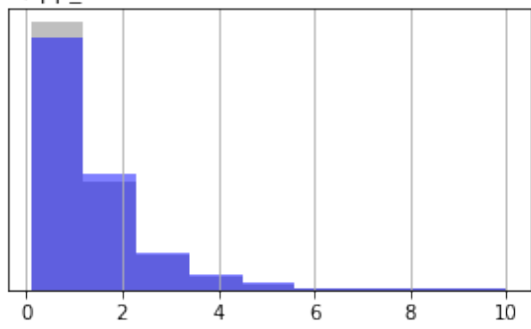
C) pp_sy2



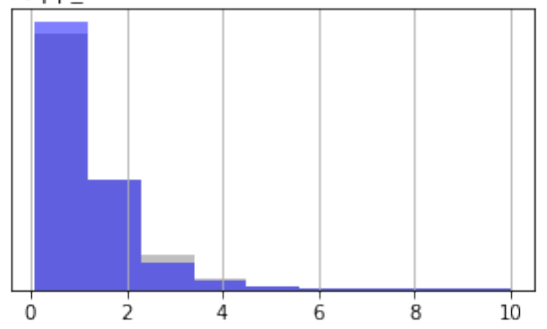
D) pp_vka0



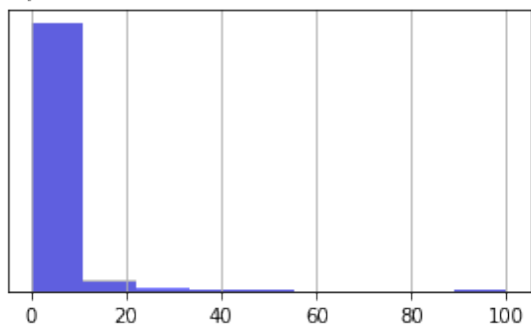
E) pp_vka1



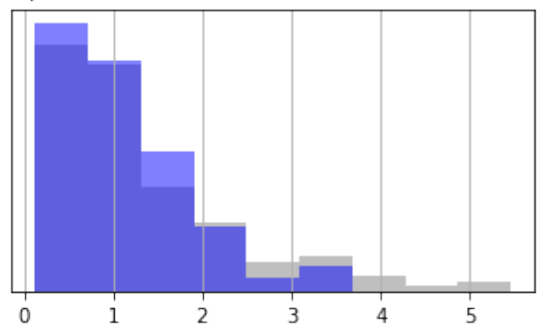
F) pp_vka2



G) strk



H) welflux



A) welflux_k02

