In [1]:

```python
import xarray as xr
import numpy as np
import geopandas as gpd
import matplotlib.pyplot as plt
import warnings
warnings.filterwarnings('ignore')
```

In [2]:

```python
plt.style.use('ggplot')
```

In [3]:

```python
gpm = xr.open_mfdataset('../2005/*')
```

In [4]:

```python
imd = xr.open_dataset("_Clim_Pred_LRF_New_RF25_IMD0p252005.nc")
```

Creating a function to correct timestamps of GPM data and to mask the data w.r.t IMD data

Also the coordinates of IMD data has been renamed to short and lower form, i.e., TIME --> time, LONGITUDE --> lon, and LATITUDE --> lat.

In [5]:

```python
def masked(gpm,imd):
    '''
    gpm: data array to be masked
    imd: maksed dataarray
    returns: masked gpm dataarray
    '''
    gpm1 = gpm.copy()
    gpm1 = gpm1.transpose("time","lat","lon")

    imd1 = imd.copy()
    imd1 = imd1.rename({'TIME':'time',"LATITUDE":'lat',"LONGITUDE":'lon'})

    gpm1 = gpm1.sel(lat = slice(imd1.lat.min(),imd1.lat.max()),
            lon = slice(imd1.lon.min(),imd1.lon.max()))
    stdTime = gpm1.indexes['time'].to_datetimeindex()
    gpm1['time'] = stdTime
    gpi = gpm1.interp_like(imd1,method='nearest')
    gp = np.ma.array(gpi, mask=np.isnan(imd1))
    gpp = xr.DataArray(gp,coords=gpi.coords)
    return gpp,imd1
```

Calling Function

In [6]:

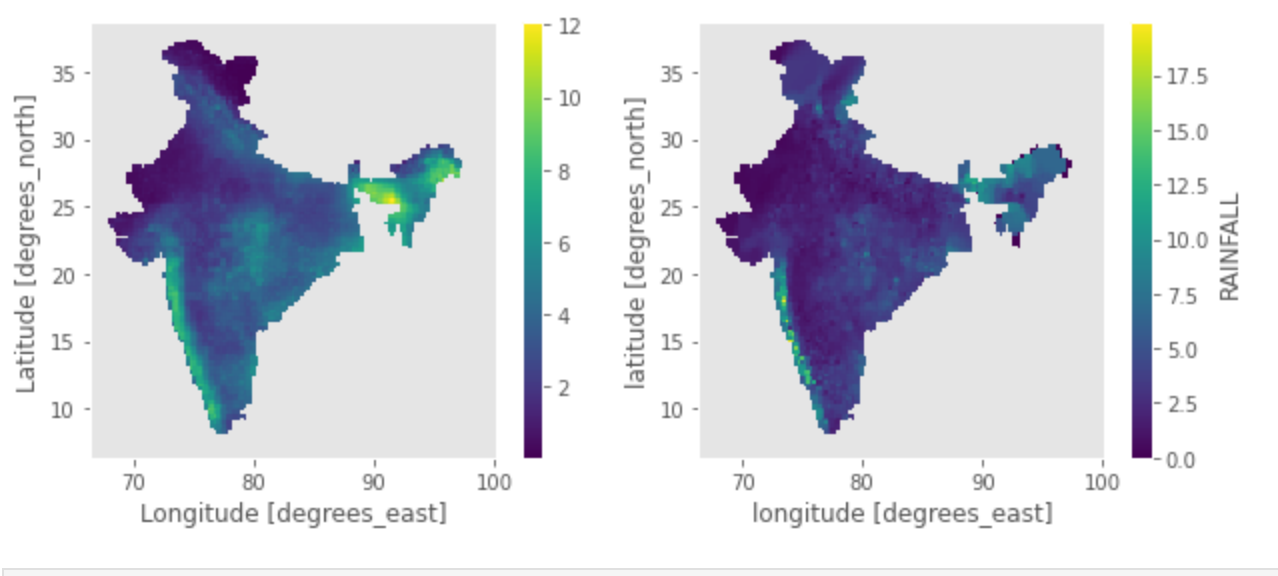
```python
gpm_new,imd_new = masked(gpm.precipitationCal,imd.RAINFALL)
```

In [7]:

```python
fig = plt.figure(figsize=[10,4])
ax = plt.subplot(121)
gpm_new.mean("time").plot()
ax = plt.subplot(122)
imd_new.mean("time").plot()
```

Out[7]:

```
<matplotlib.collections.QuadMesh at 0x2aab87902340>
```

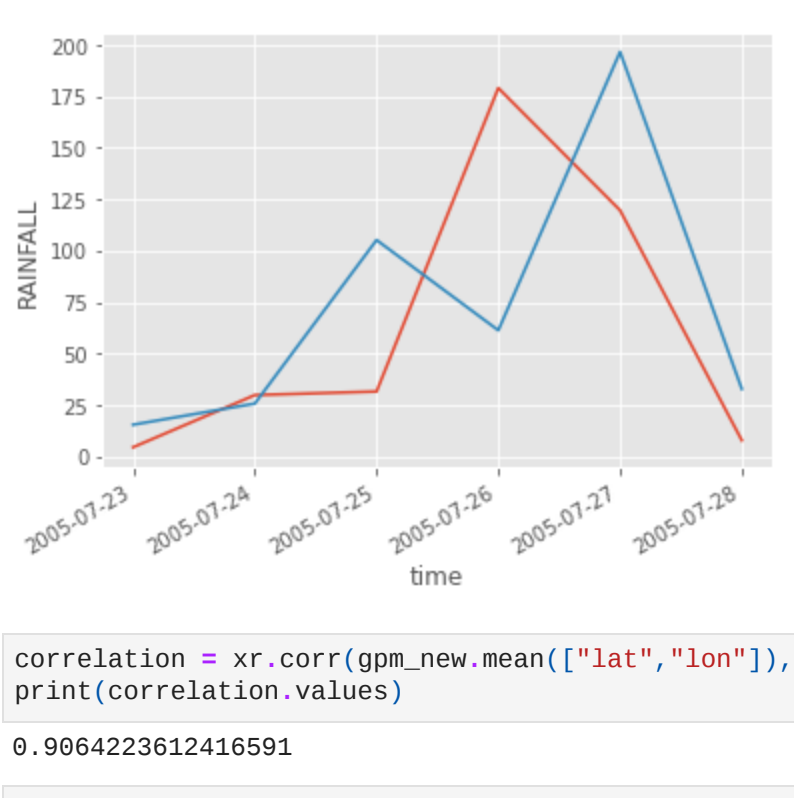


In [8]:

```python
gpm_new.mean(["lat","lon"]).plot()
imd_new.mean(["lat","lon"]).plot()
```

Out[8]:

```
[<matplotlib.lines.Line2D at 0x2aab88931340>]
```

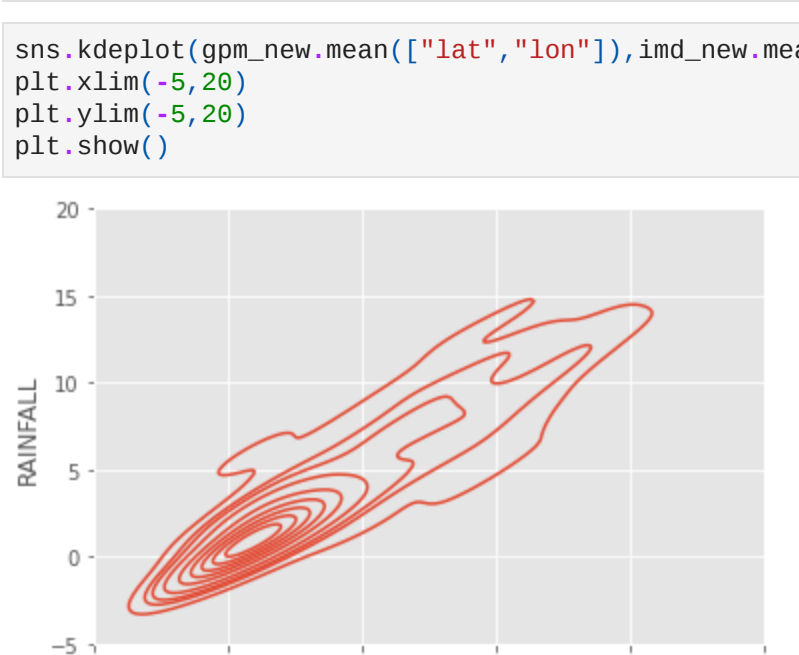


In [9]:

```python
gpm_new.sel(time=slice('2005-07-23','2005-07-28'),lat=slice(17,20),lon=slice(71,73)).mean(["lon","lat"]).plot()
imd_new.sel(time=slice('2005-07-23','2005-07-28'),lat=slice(17,20),lon=slice(71,73)).mean(["lon","lat"]).plot()
```

Out[9]:

```
[<matplotlib.lines.Line2D at 0x2aab88bced60>]
```



In [10]:

```python
correlation = xr.corr(gpm_new.mean(["lat","lon"]),imd_new.mean(["lat","lon"]))
print(correlation.values)
```

```
0.9064223612416591
```

In [11]:

```python
bias = gpm_new.mean().values-imd_new.mean().values
print(bias)
```

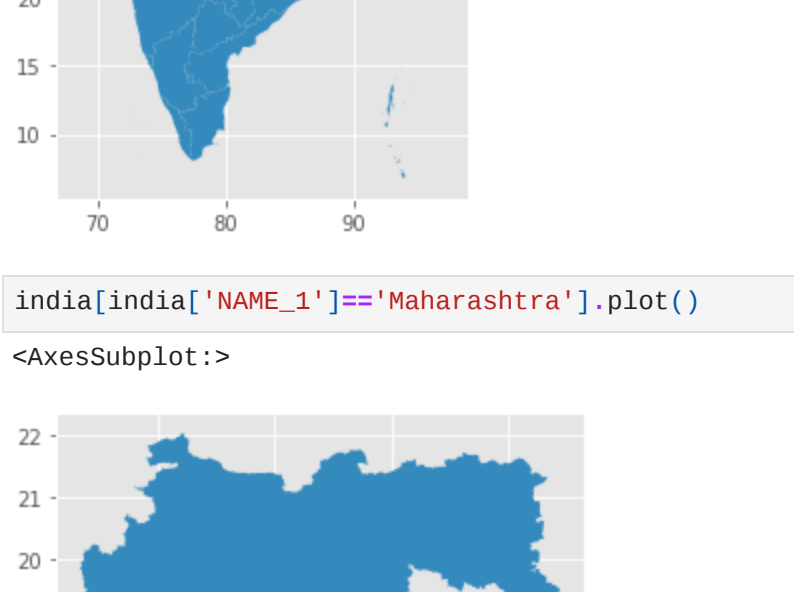
```
0.08560181
```

In [12]:

```python
import seaborn as sns
```

In [13]:

```python
sns.kdeplot(gpm_new.mean(["lat","lon"]),imd_new.mean(["lat","lon"]))
plt.xlim(-5,20)
plt.ylim(-5,20)
plt.show()
```

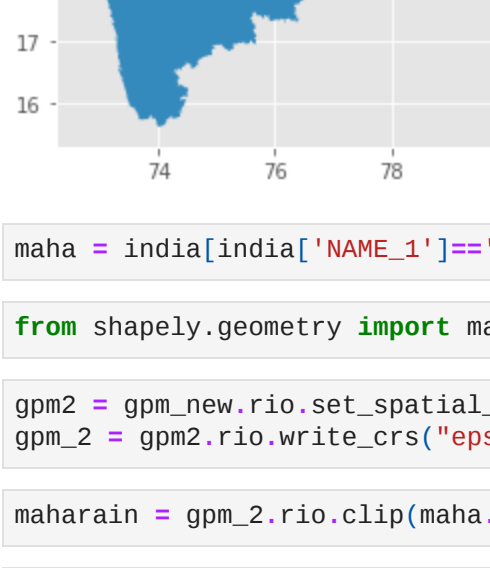


In [14]:

```python
india = gpd.read_file('IND_adm1.shp')
india.plot()
```

Out[14]:

```
<AxesSubplot:>
```

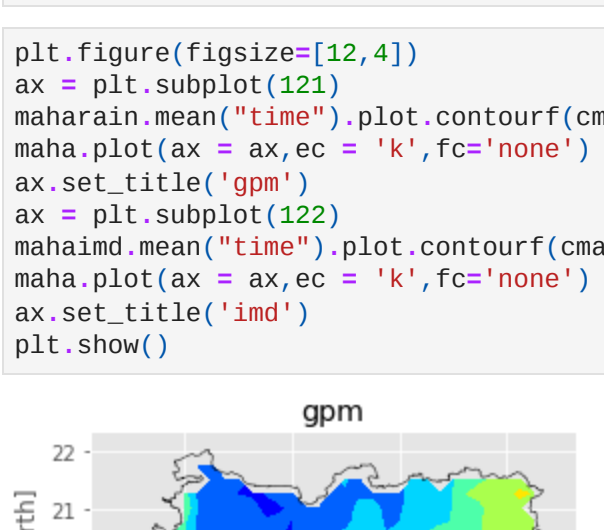


In [15]:

```python
india[india['NAME_1']=='Maharashtra'].plot()
```

Out[15]:

```
<AxesSubplot:>
```



In [16]:

```python
maha = india[india['NAME_1']=='Maharashtra']
```

In [17]:

```python
from shapely.geometry import mapping
```

In [18]:

```python
gpm2 = gpm_new.rio.set_spatial_dims(x_dim="lon", y_dim="lat", inplace=False)
gpm_2 = gpm2.rio.write_crs("epsg:4326")
```

In [19]:

```python
maharain = gpm_2.rio.clip(maha.geometry.apply(mapping),crs=maha.crs)
```

In [20]:

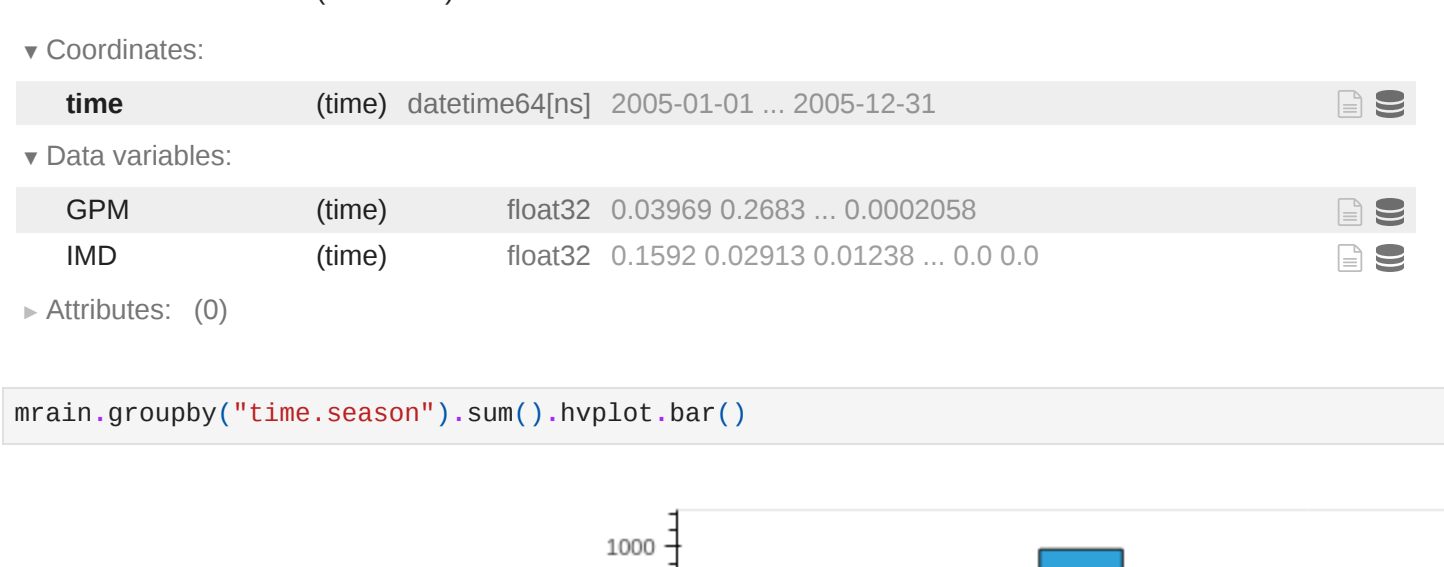
```python
imd_2 = imd_new.rio.write_crs("epsg:4326")
```

In [21]:

```python
mahaimd = imd_2.rio.clip(maha.geometry.apply(mapping),maha.crs)
```

In [22]:

```python
plt.figure(figsize=[12,4])
ax = plt.subplot(121)
maharain.mean("time").plot.contourf(cmap='jet',levels=range(0,10),extend="max")
maha.plot(ax = ax,ec = 'k',fc='none')
ax.set_title('gpm')
ax = plt.subplot(122)
mahaimd.mean("time").plot.contourf(cmap='jet',levels=range(0,10),extend='max')
maha.plot(ax = ax,ec = 'k',fc='none')
ax.set_title('imd')
plt.show()
```



In [25]:

```python
import hvplot.xarray
```

In [109…

```python
mrain = xr.Dataset({"GPM":maharain.mean(["lat","lon"]),
            "IMD":mahaimd.mean(["lat","lon"])}).drop({"spatial_ref"})
mrain
```

Out[109]:

xarray.Dataset

- Dimensions: (**time**: 365)
- Coordinates:

time	(time)	datetime64[ns]	2005-01-01 ... 2005-12-31

- Data variables:

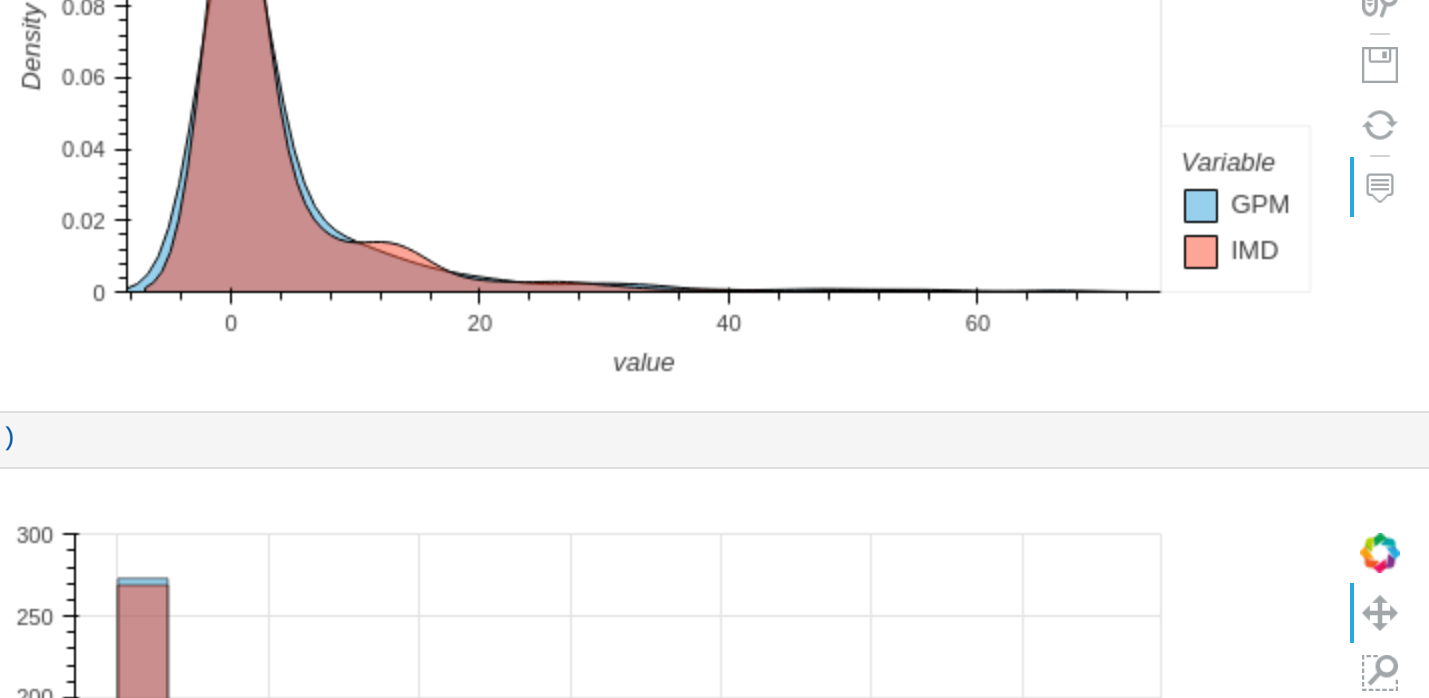
GPM	(time)	float32	0.03969 0.2683 ... 0.0002058
IMD	(time)	float32	0.1592 0.02913 0.01238 ... 0.0 0.0

- Attributes: (0)

In [77]:

```python
mrain.groupby("time.season").sum().hvplot.bar()
```

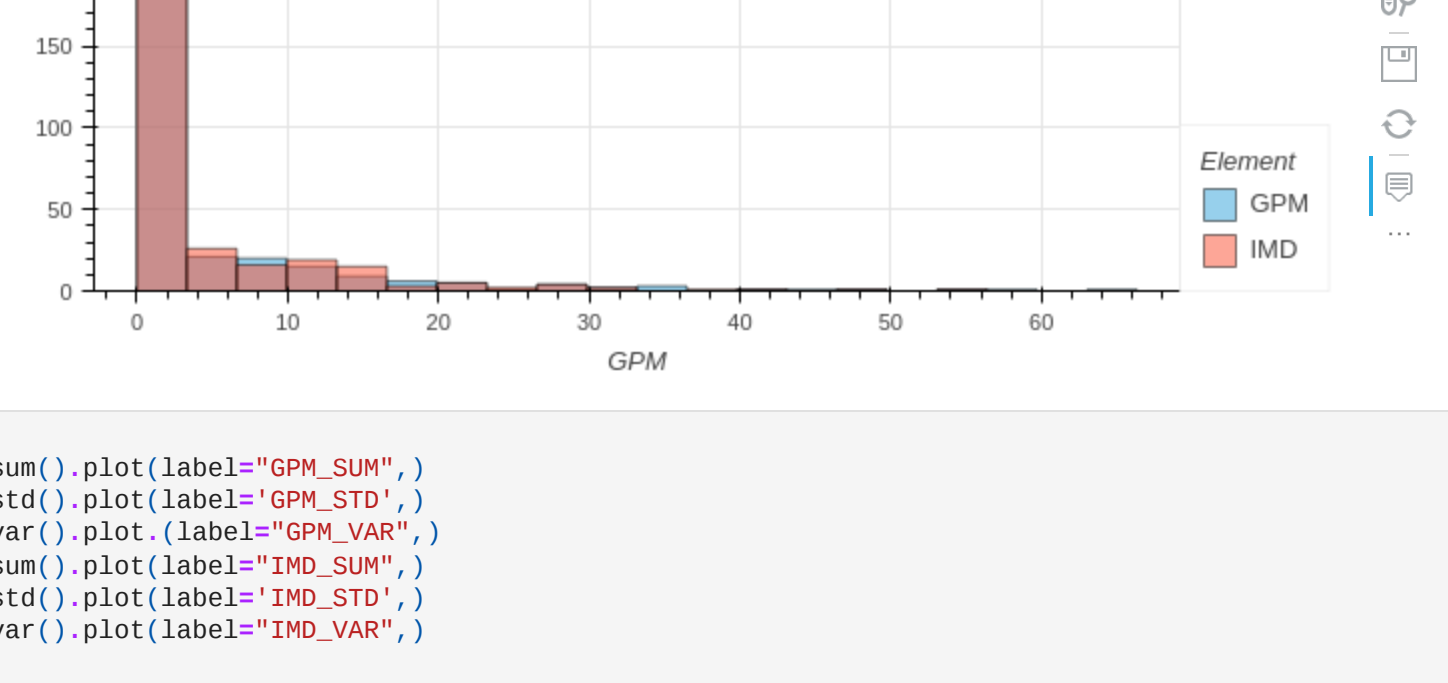
Out[77]:



In [61]:

```python
mrain.hvplot.density()
```

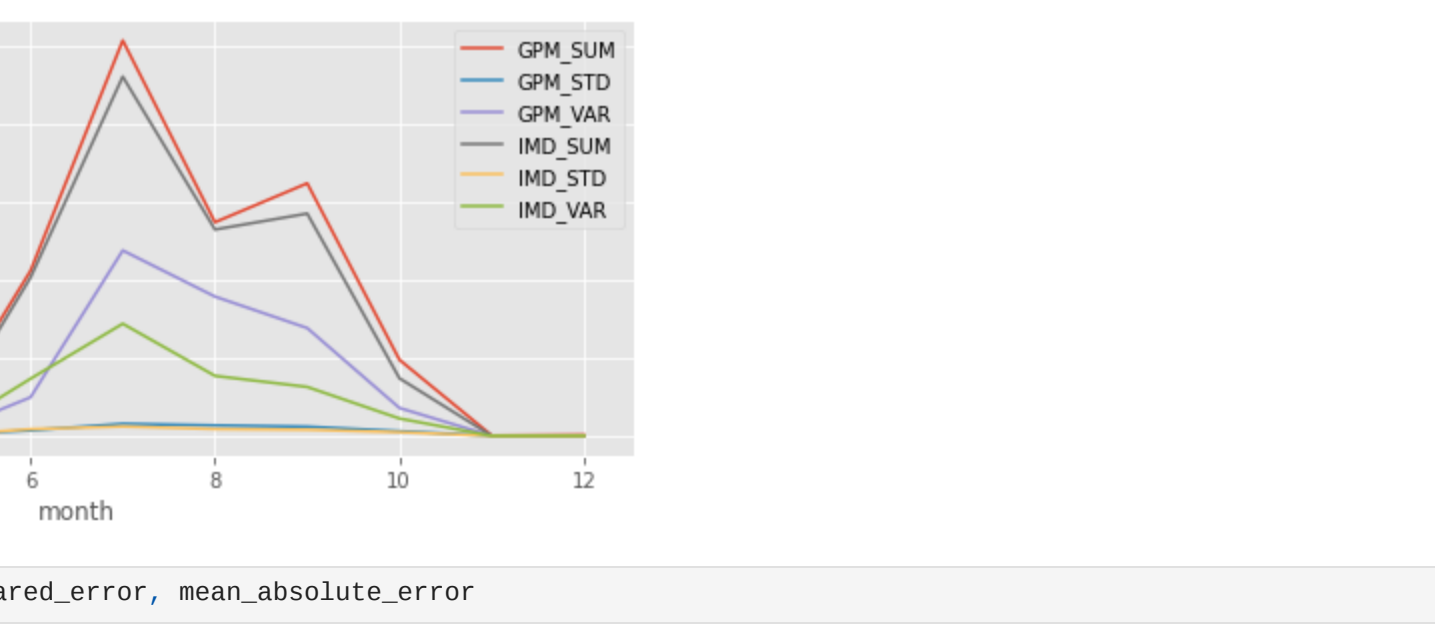
Out[61]:



In [65]:

```python
mrain.hvplot.hist(alpha=0.5,grid=True)
```

Out[65]:

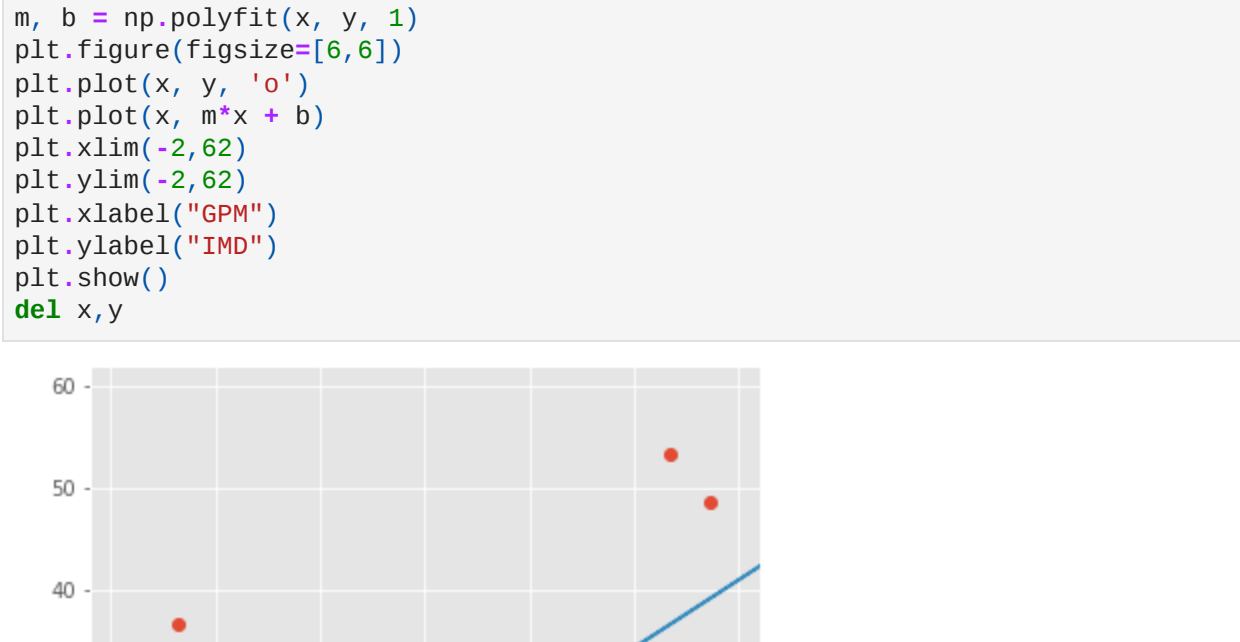


In [111…

```python
plt.figure(figsize=(10,4))
mrain["GPM"].groupby("time.month").sum().plot(label="GPM_SUM",)
mrain["GPM"].groupby("time.month").std().plot(label='GPM_STD',)
mrain["GPM"].groupby("time.month").var().plot.(label="GPM_VAR",)
mrain["IMD"].groupby("time.month").sum().plot(label="IMD_SUM",)
mrain["IMD"].groupby("time.month").std().plot(label='IMD_STD',)
mrain["IMD"].groupby("time.month").var().plot(label="IMD_VAR",)
plt.legend()
```

Out[111]:

```
<matplotlib.legend.Legend at 0x2aad05977eb0>
```



In [335…

```python
from sklearn.metrics import mean_squared_error, mean_absolute_error
```

In [336…

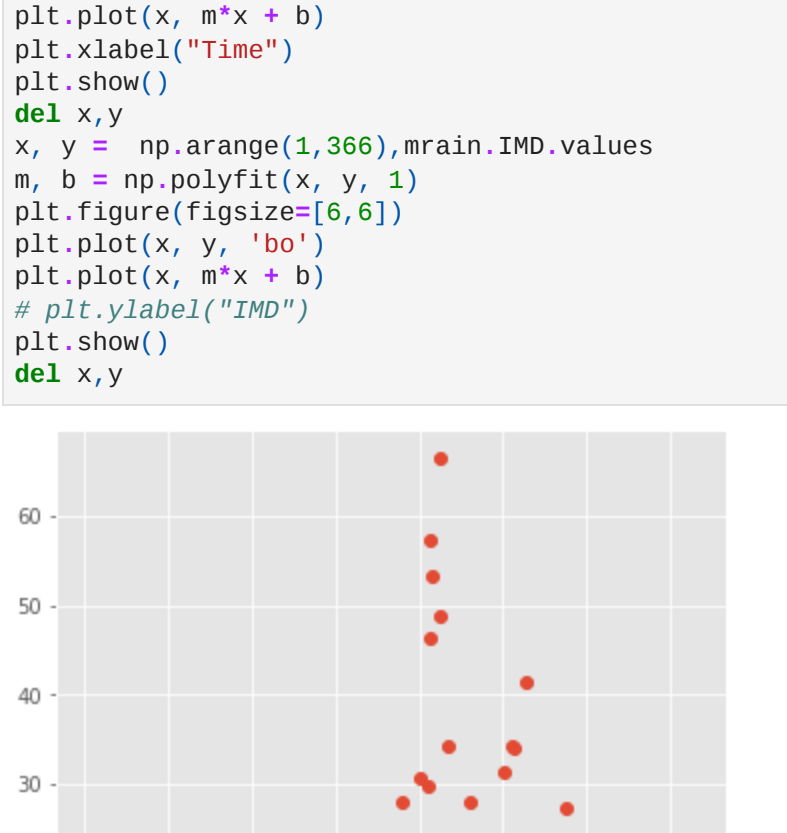
```python
print("RMSE", mean_squared_error(mrain.GPM,mrain.IMD))
print("MAE",mean_absolute_error(mrain.GPM,mrain.IMD))
```

```
RMSE 28.681728
MAE 2.3130205
```

In [338…

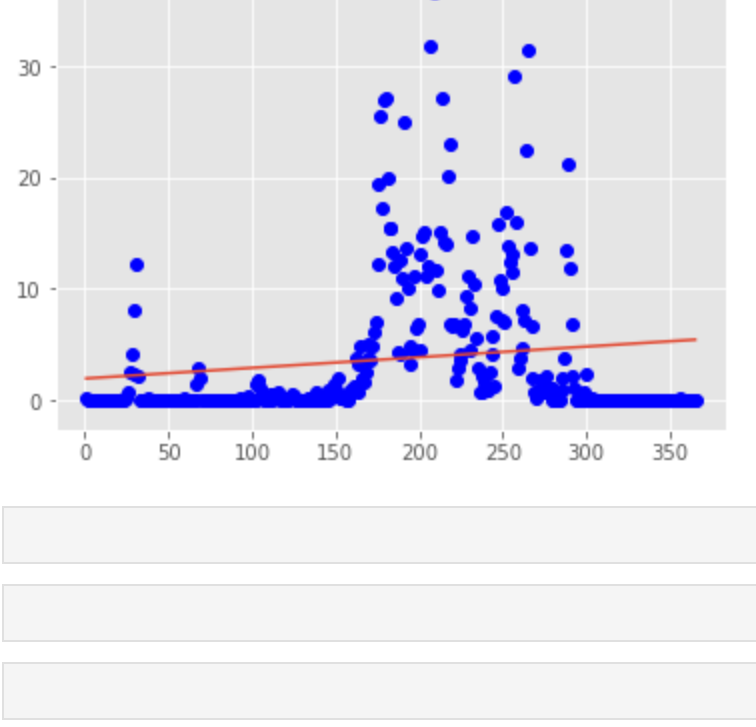
In [354…

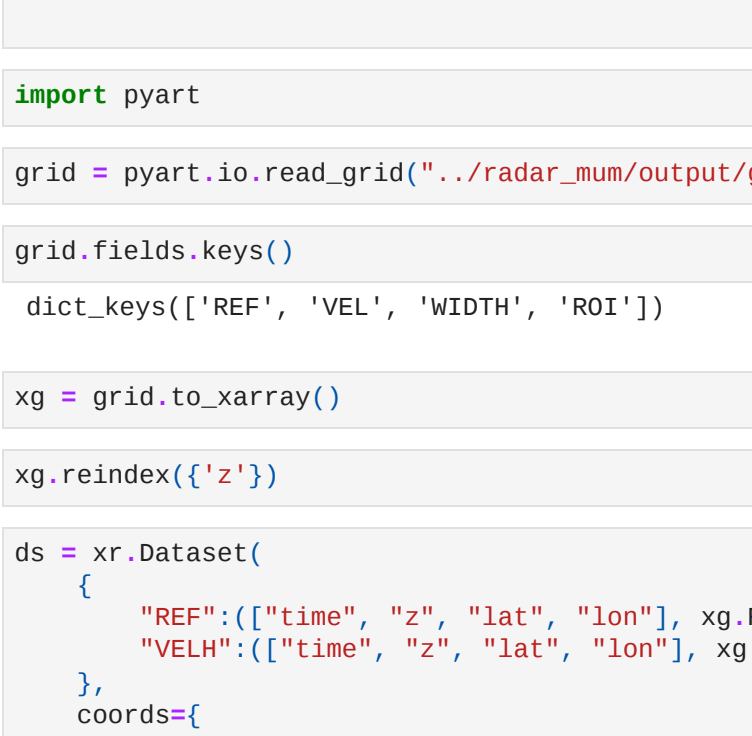
```python
x, y =  mrain.GPM,mrain.IMD
m, b = np.polyfit(x, y, 1)
plt.figure(figsize=[6,6])
plt.plot(x, y, 'o')
plt.plot(x, m*x + b)
plt.xlim(-2,62)
plt.ylim(-2,62)
plt.xlabel("GPM")
plt.ylabel("IMD")
plt.show()
del x,y
```



In [393…

```python
x, y =  np.arange(1,366),mrain.GPM.values
m, b = np.polyfit(x, y, 1)
plt.figure(figsize=[6,6])
ax = plt.axes()
plt.plot(x, y, 'o')
plt.plot(x, m*x + b)
plt.xlabel("Time")
plt.show()
del x,y
x, y =  np.arange(1,366),mrain.IMD.values
m, b = np.polyfit(x, y, 1)
plt.figure(figsize=[6,6])
plt.plot(x, y, 'bo')
plt.plot(x, m*x + b)
# plt.ylabel("IMD")
plt.show()
del x,y
```





In [ ]:

In [ ]:

In [ ]:

In [ ]:

In [255…

```python
import pyart
```

In [256…

```python
grid = pyart.io.read_grid("../radar_mum/output/grid_MUM150615000342.nc")
```

In [258…

```python
grid.fields.keys()
```

Out[258]:

```
dict_keys(['REF', 'VEL', 'WIDTH', 'ROI'])
```

In [259…

```python
xg = grid.to_xarray()
```

In [278…

```python
xg.reindex({'z'})
```

In [321…

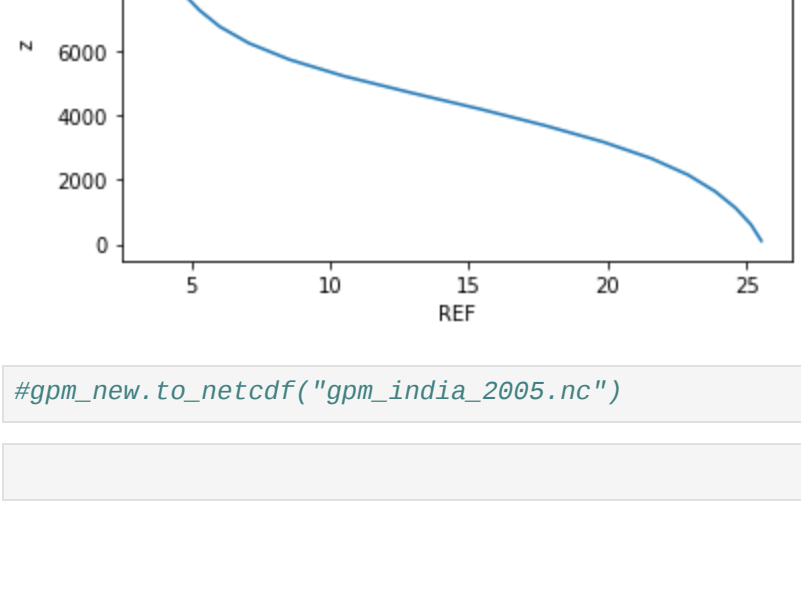
```python
ds = xr.Dataset(
    {
        "REF":(["time", "z", "lat", "lon"], xg.REF.values,),
        "VELH":(["time", "z", "lat", "lon"], xg.VEL.values,),
    },
    coords={
        "z" :(['z'], xg.z.values),
        "lon" :(["lon"], xg.lon.values),
        "lat" :(["lat"], xg.lat.values),
        "time" :(["time"], xg.time.values),
    }
)
```

In [338…

```python
ds['REF'][0,4].plot(cmap='pyart_NWSRef',vmin=-10,vmax=60)
```

Out[338]:

```
<matplotlib.collections.QuadMesh at 0x2aad24e31430>
```



In [344…

```python
ds["REF"][0].sel(lat=19,lon=71.5,method='nearest').plot(y='z')
```

Out[344]:

```
[<matplotlib.lines.Line2D at 0x2aad24f93af0>]
```


In [337…

```python
#gpm_new.to_netcdf("gpm_india_2005.nc")
```

In [ ]: