In [170…
```
import pandas as pd
import xarray as xr
import netCDF4 as nc
import numpy as np
import cartopy.crs as ccrs
import matplotlib.pyplot as plt
import cartopy.feature as cfeature
```

In [64]:
```
data= xr.open_mfdataset('*.nc')
data
```

Out[64]: xarray.Dataset

- ▸ Dimensions: (**lon**: 94, **lat**: 87, **datetime**: 3652)
- ▾ Coordinates:

lon	(lon)	float32	122.8 123.0 123.2 ... 1…
lat	(lat)	float32	46.0 45.75 45.5 ... 25.…
datetime	(datetime)	datetime64[ns]	2001-01-01 ... 2010-12…

- ▾ Data variables:

crs	(datetime)	int32	-2147483647 ... -2147…
precip	(datetime, lat, lon)	float32	dask.array<chunksize…

- ▸ Indexes: (3)
- ▾ Attributes:

Conventions :	CF-1.6

In [65]:
```
data=data.where(data !=-99.0)
```

In [66]:
```
data
```

Out[66]: xarray.Dataset

▸ Dimensions: (**datetime**: 3652, **lat**: 87, **lon**: 94)

▾ Coordinates:

lon	(lon)	float32	122.8 123.0 123.2 ... 1…
lat	(lat)	float32	46.0 45.75 45.5 ... 25.…
datetime	(datetime)	datetime64[ns]	2001-01-01 ... 2010-12…

▾ Data variables:

crs	(datetime)	float64	-2.147e+09 ... -2.147e…
precip	(datetime, lat, lon)	float32	dask.array<chunksize…

▸ Indexes: (3)

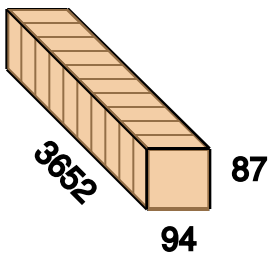
▾ Attributes:

Conventions :	CF-1.6

In [67]:
```
japan_rr=data.precip
japan_rr
```

Out[67]: xarray.DataArray 'precip' (**datetime**: 3652, **lat**: 87, **lon**: 94)

	Array	Chunk
Bytes	113.93 MiB	11.42 MiB
Shape	(3652, 87, 94)	(366, 87, 94)
Count	50 Tasks	10 Chunks
Type	float32	numpy.ndarray



▾ Coordinates:

lon	(lon)	float32	122.8 123.0 123.2 ... 145.8 14…
lat	(lat)	float32	46.0 45.75 45.5 ... 25.0 24.75 …
datetime	(datetime)	datetime64[ns]	2001-01-01 ... 2010-12-31

▸ Indexes: (3)

▾ Attributes:

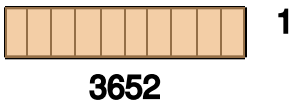
grid_mapping :	crs

In [68]:
```
japan_RR=japan_rr.sel(lon=138.25, lat=36.20, method='nearest')
```

In [69]:
```
japan_RR
```

Out[69]: xarray.DataArray 'precip' (**datetime**: 3652)

	Array	Chunk
Bytes	14.27 kiB	1.43 kiB
Shape	(3652,)	(366,)
Count	60 Tasks	10 Chunks
Type	float32	numpy.ndarray



▾ Coordinates:

lon	()	float32	138.2	
lat	()	float32	36.25	
datetime	(datetime)	datetime64[ns]	2001-01-01 ... 2010-12-31	

▸ Indexes: (1)

▾ Attributes:

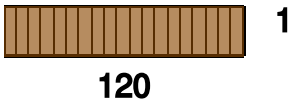
grid_mapping : crs

In [ ]:

In [70]:
```
monthlyrr_total=japan_RR.resample(datetime='1M').sum('datetime')
monthlyrr_total
```

Out[70]: xarray.DataArray 'precip' (**datetime**: 120)

	Array	Chunk
Bytes	480 B	4 B
Shape	(120,)	(1,)
Count	784 Tasks	120 Chunks
Type	float32	numpy.ndarray



▾ Coordinates:

lon	()	float32	138.2	
lat	()	float32	36.25	
datetime	(datetime)	datetime64[ns]	2001-01-31 ... 2010-12-31	

▸ Indexes: (1)

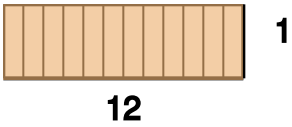
▾ Attributes:

grid_mapping : crs

In [71]:
```
monthhthly_climo=monthlyrr_total.groupby('datetime.month').mean('datetime')
monthhthly_climo
```

Out[71]: xarray.DataArray 'precip' (**month**: 12)

	Array	Chunk
Bytes	48 B	4 B
Shape	(12,)	(1,)
Count	1084 Tasks	12 Chunks
Type	float32	numpy.ndarray



▾ Coordinates:

lon	()	float32	138.2	
lat	()	float32	36.25	
month	(month)	int64	1 2 3 4 5 6 7 8 9 10 11 12	

▸ Indexes: (1)

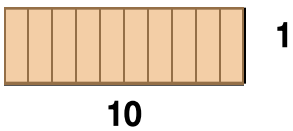
▾ Attributes:

grid_mapping : crs

In [72]:
```
annualrr_total=japan_RR.resample(datetime='1Y').sum('datetime')
annualrr_total
```

Out[72]: xarray.DataArray 'precip' (**datetime**: 10)

	Array	Chunk
Bytes	40 B	4 B
Shape	(10,)	(1,)
Count	122 Tasks	10 Chunks
Type	float32	numpy.ndarray



▾ Coordinates:

lon	()	float32	138.2	
lat	()	float32	36.25	
datetime	(datetime)	datetime64[ns]	2001-12-31 ... 2010-12-31	

▸ Indexes: (1)

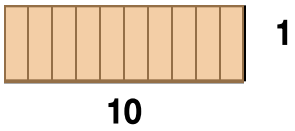
▾ Attributes:

grid_mapping : crs

In [73]:
```
annual_total=japan_rr.groupby('datetime.year').sum()
annual_total

annual_clime=annual_total.mean(dim=['lat','lon'])
annual_clime
```

Out[73]: xarray.DataArray 'precip' (**year**: 10)

	Array	Chunk
Bytes	40 B	4 B
Shape	(10,)	(1,)
Count	132 Tasks	10 Chunks
Type	float32	numpy.ndarray



▾ Coordinates:

year (year) int64 2001 2002 2003 ... 2008 2009 2010

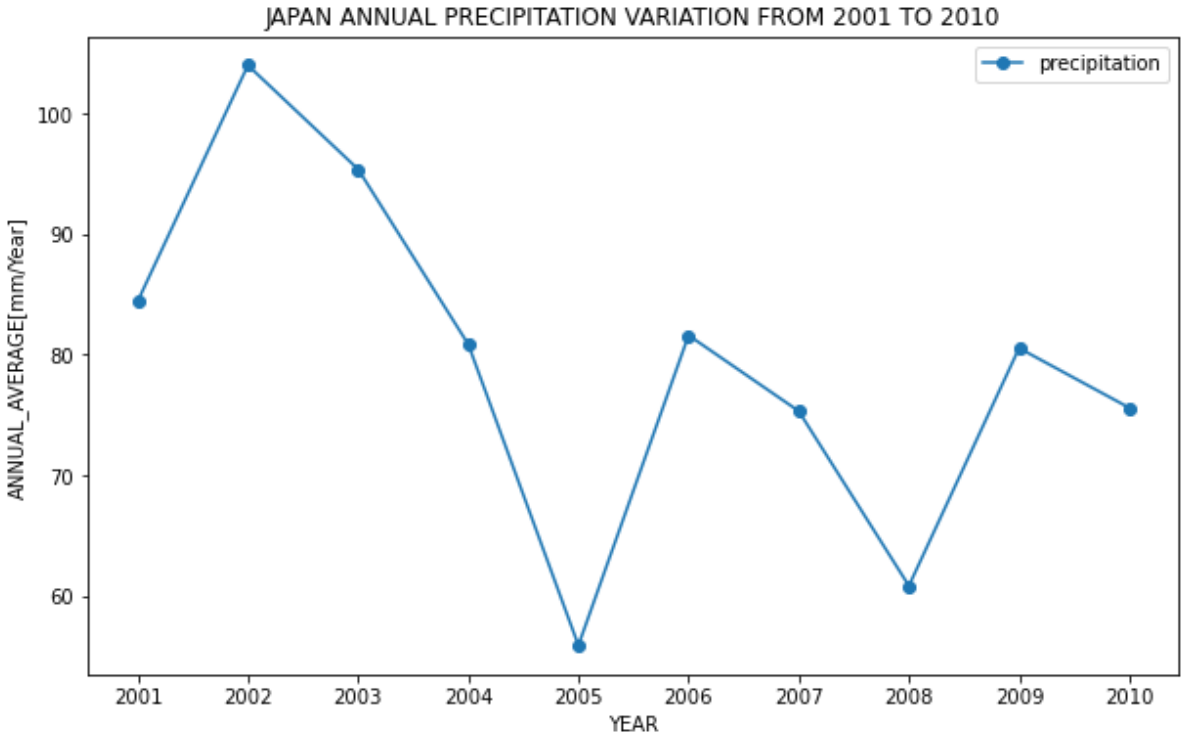
▸ Indexes: (1)

▸ Attributes: (0)

In [74]:
```
month=['Jan','Feb','Mar','Apr','May','Jun','Jul','Aug','Sept','Oct','Nov','D
```

In [75]:
```
year=['2001','2002','2003','2004','2005','2006','2007','2008','2009','2010']
```

In [76]:
```
plt.subplots(figsize=(10,6))
plt.plot(year,annual_clime,marker='o', label='precipitation')
plt.xlabel('YEAR')
plt.ylabel('ANNUAL_AVERAGE[mm/Year]')
plt.title('JAPAN ANNUAL PRECIPITATION VARIATION FROM 2001 TO 2010')
plt.legend(loc='upper right')
plt.savefig('annual average.png')
```



In [77]:
```
dryDays_per_year = japan_RR.where(japan_RR < 1).groupby('datetime.year').cou
```

In [78]:
```
wetDays_per_year = japan_RR.where(japan_RR >= 1).groupby('datetime.year').co
```

In [79]:
```
dryDays_per_month = japan_RR.where(japan_RR < 1).groupby('datetime.month').c
```

In [80]:
```
wetDays_per_month = japan_RR.where(japan_RR >= 1).groupby('datetime.month').
```

In [81]:
```
xrain10 = japan_RR.where(japan_RR > 10).groupby('datetime.year').count('date
```

In [82]:
```
xrain20 = japan_RR.where(japan_RR > 20).groupby('datetime.year').count('date
```

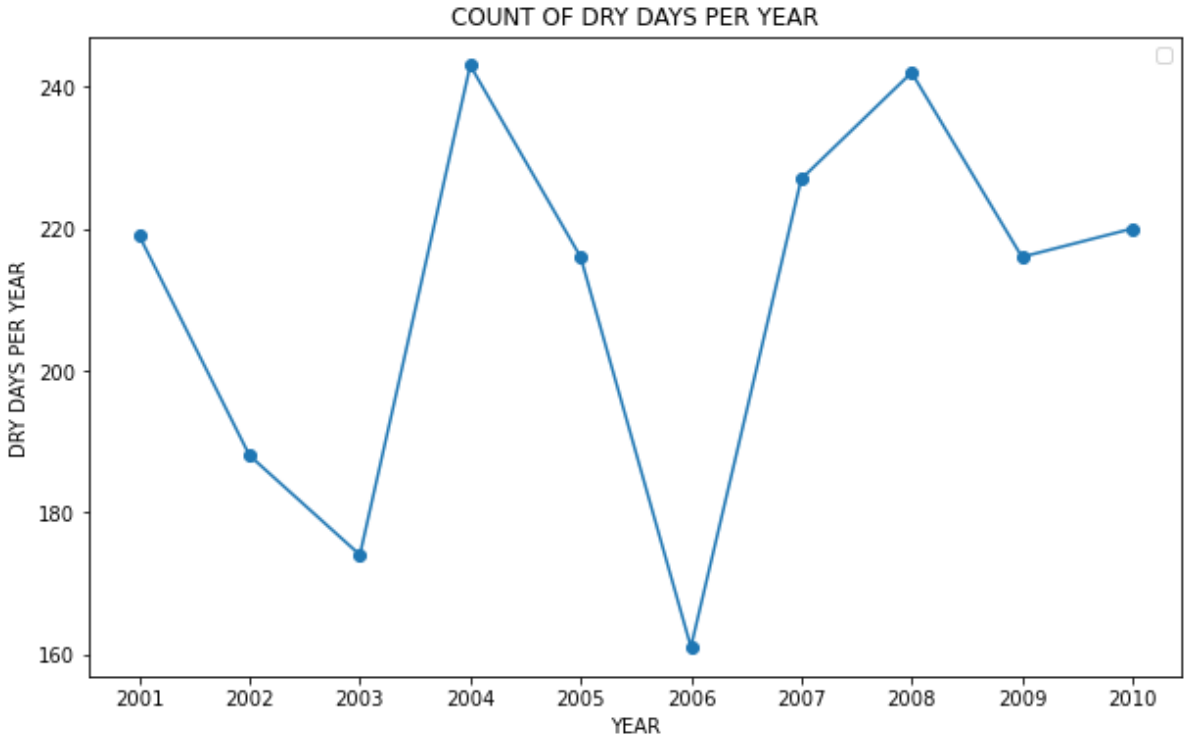
In [83]:
```
xrain10M = japan_RR.where((japan_RR > 10) & (japan_RR <=20)).groupby('dateti
```

In [84]:
```
xrain20M = japan_RR.where(japan_RR > 20).groupby('datetime.month').count('da
```

In [85]:
```
#
```

In [157…
```
#plotting time series for drydays per year
fig,ax = plt.subplots(figsize=(10,6))
#plt.sublots_adjust(hspace=0.5, wspace=0.2)
plt.plot(year,dryDays_per_year, marker='o' )
#plt.plot(year,annual_clime,marker='o', label='precipitation')
plt.xlabel('YEAR')
plt.ylabel('DRY DAYS PER YEAR')
plt.title('COUNT OF DRY DAYS PER YEAR')
plt.legend(loc='upper right')
plt.savefig('timeseries dry days per year')
```

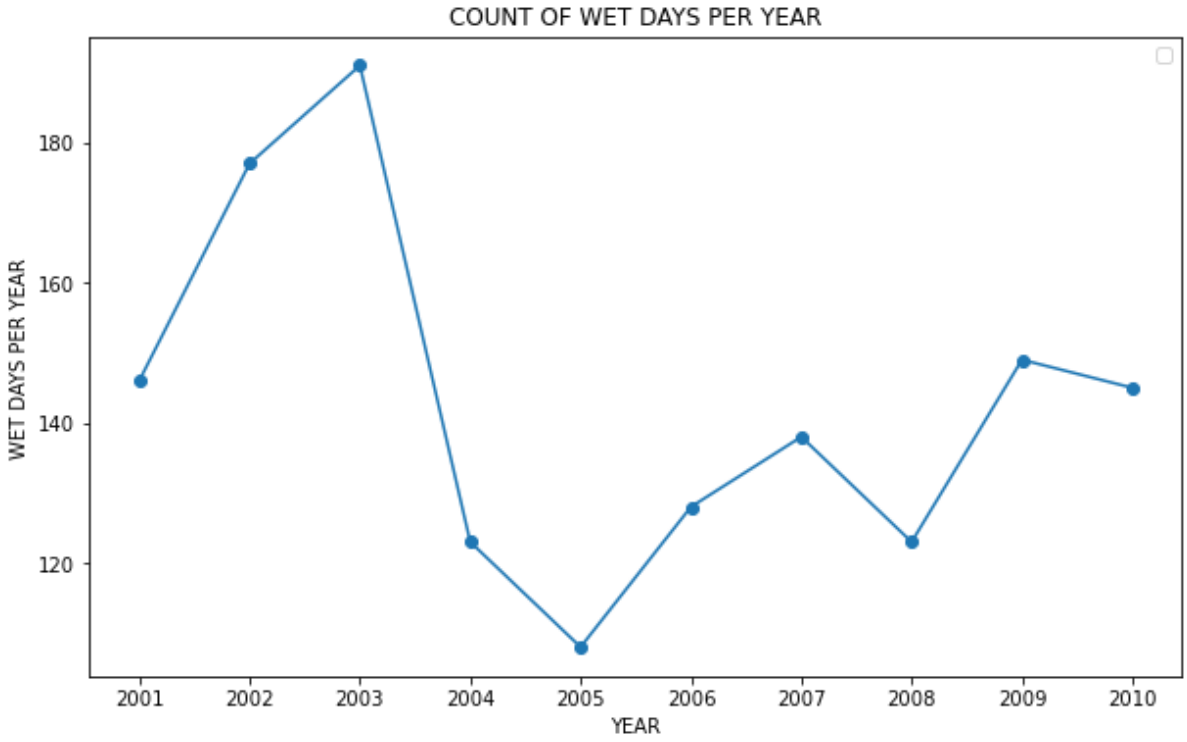
```
No handles with labels found to put in legend.
```



In [151…

```
#plotting time series for wetdays per year
fig,ax = plt.subplots(figsize=(10,6))
#plt.sublots_adjust(hspace=0.5, wspace=0.2)
plt.plot(year,wetDays_per_year, marker='o' )
#plt.plot(year,annual_clime,marker='o', label='precipitation')
plt.xlabel('YEAR')
plt.ylabel('WET DAYS PER YEAR')
plt.title('COUNT OF WET DAYS PER YEAR')
plt.legend(loc='upper right')
plt.savefig('timeseries wet days per year')
```

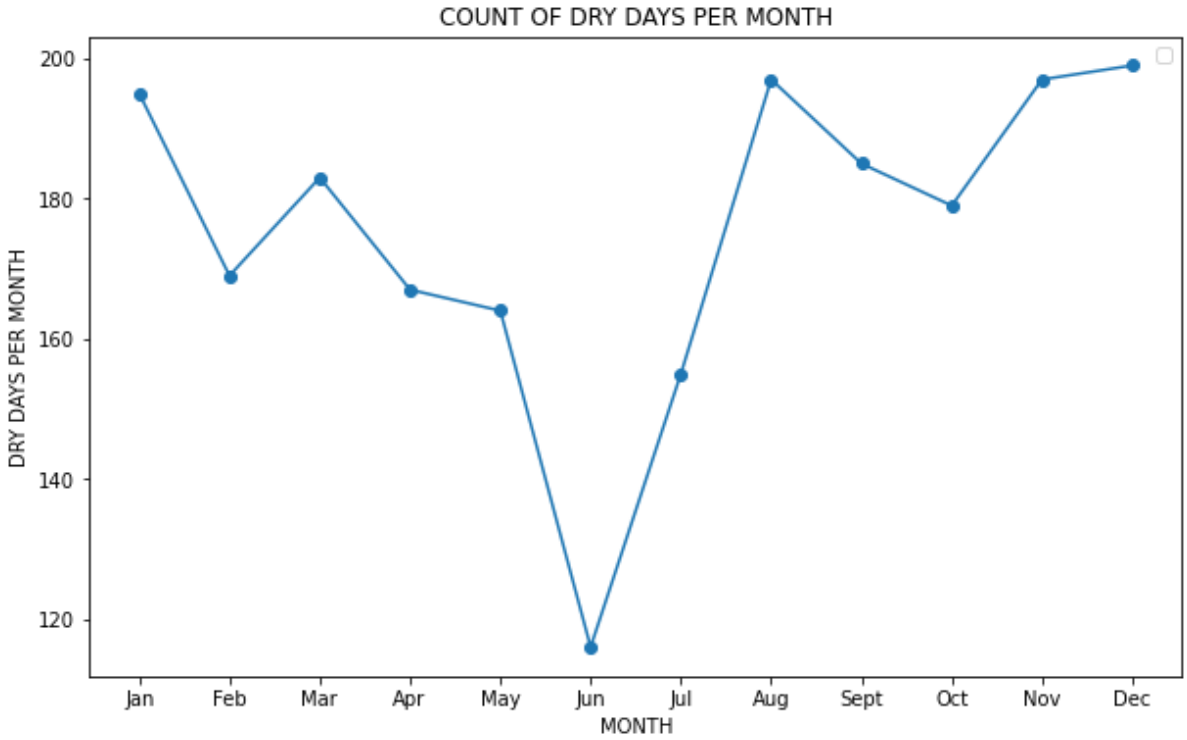
```
No handles with labels found to put in legend.
```



In [145…

```
#plotting time series for drydays per month
fig,ax = plt.subplots(figsize=(10,6))
#plt.sublots_adjust(hspace=0.5, wspace=0.2)
plt.plot(month,dryDays_per_month, marker='o' )
#plt.plot(year,annual_clime,marker='o', label='precipitation')
plt.xlabel('MONTH')
plt.ylabel('DRY DAYS PER MONTH')
plt.title('COUNT OF DRY DAYS PER MONTH')
plt.legend(loc='upper right')
plt.savefig('timeseries of dry days per month')
```

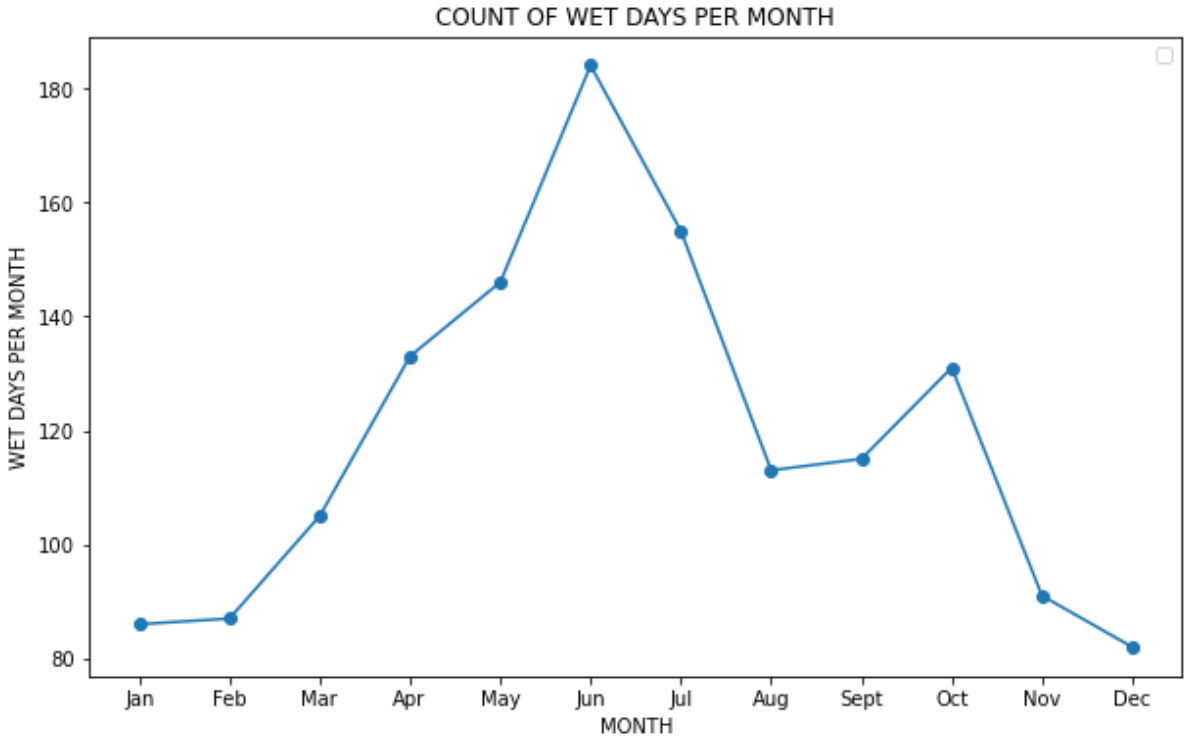
```
No handles with labels found to put in legend.
```



In [142…

```
#plotting time series for wetdays per year
fig,ax = plt.subplots(figsize=(10,6))
#plt.sublots_adjust(hspace=0.5, wspace=0.2)
plt.plot(month,wetDays_per_month, marker='o' )
#plt.plot(year,annual_clime,marker='o', label='precipitation')
plt.xlabel('MONTH')
plt.ylabel('WET DAYS PER MONTH')
plt.title('COUNT OF WET DAYS PER MONTH')
plt.legend(loc='upper right')
plt.savefig('timeseries of wet days per month')
```

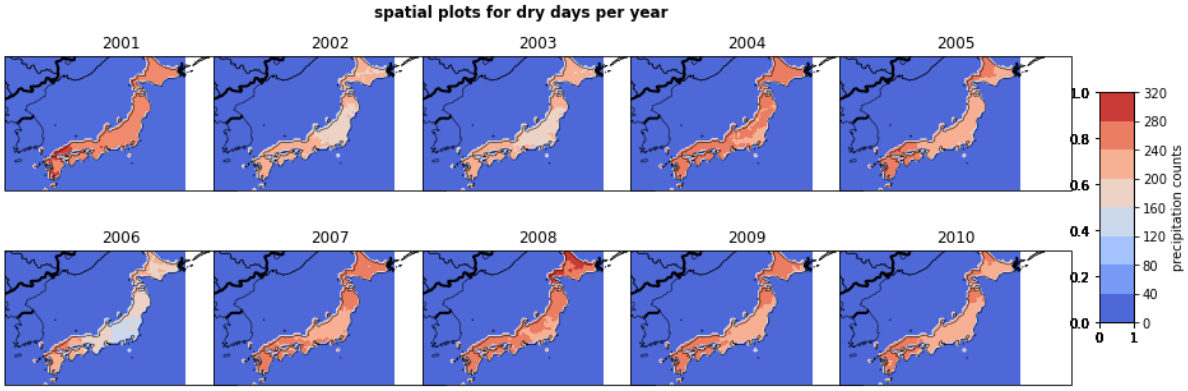
```
No handles with labels found to put in legend.
```



SPATIAL PLOTS

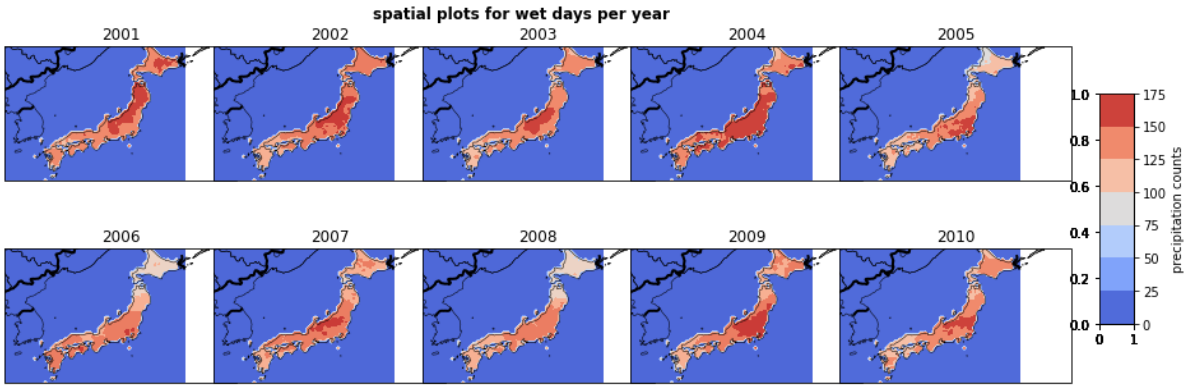
In [92]:
```
dryDays_per_year_ts = japan_rr.where(japan_rr < 1).groupby('datetime.year').
wetDays_per_year_ts = japan_rr.where(japan_rr >= 1).groupby('datetime.year')
dryDays_per_month_ts = japan_rr.where(japan_rr < 1).groupby('datetime.month'
wetDays_per_month_ts = japan_rr.where(japan_rr >= 1).groupby('datetime.month
```

In [156…
```
#spatial plots for dry days per year
fig,ax=plt.subplots(2,5,figsize=(15,5),
                    subplot_kw={'projection': ccrs.PlateCarree()})
ax=ax.flatten()
year_names=['2001','2002','2003','2004','2005','2006','2007','2008','2009','
for i in range(10):
    ax[i].add_feature(cfeature.COASTLINE,linewidth=0.5)
    ax[i].add_feature(cfeature.BORDERS,linewidth=2)
    ax[i].add_feature(cfeature.STATES,linewidth=0.5)
     # ax[i].add_feature(cfeature.OCEAN,linewidth=2)
       # ax[i].add_feature(cfeature.LAKES,linewidth=2 color='blue')
        #    ax[i].add_feature(cfeature.RIVERS)
    ax[i].set_extent([125,152,30,45])

    ax[i].set_title(year_names[i])
    cb=ax[i].contourf(dryDays_per_year_ts.lon,dryDays_per_year_ts.lat,dryDay
    color_bar=fig.add_axes([0.92,0.29,0.025,0.5])
fig.colorbar(cb,cax=color_bar,label='precipitation counts')
plt.subplots_adjust(wspace=-0.10,top=0.9)
plt.suptitle('spatial plots for dry days per year',fontweight='bold')
plt.savefig('dry days per year.png')
```



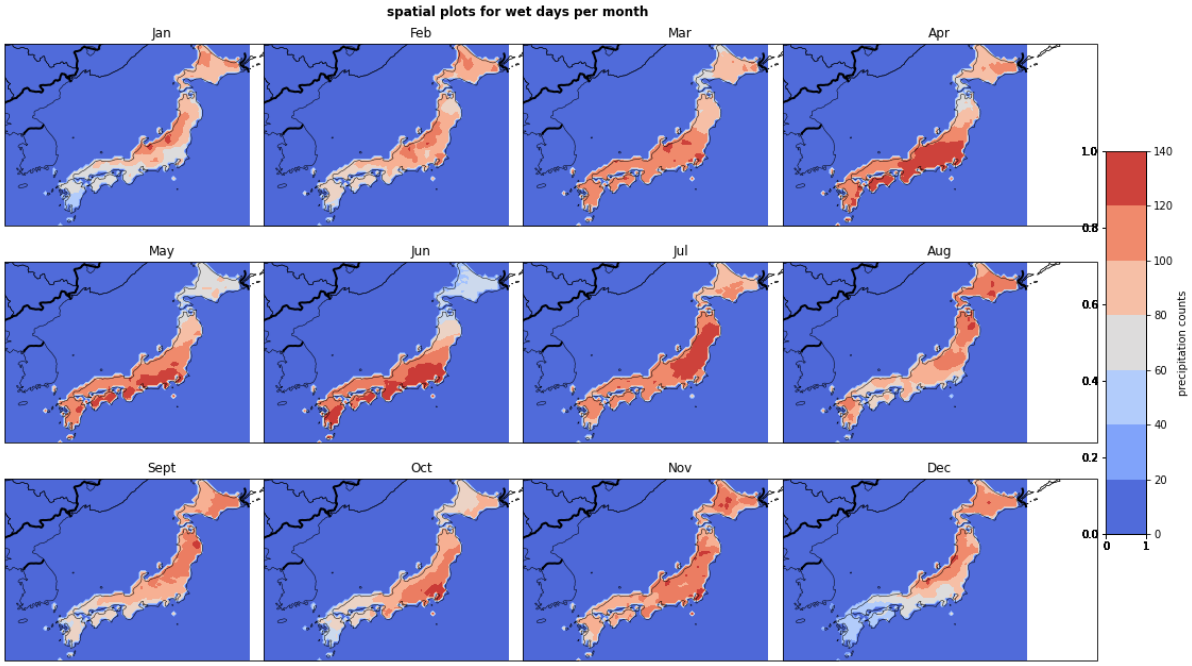
In [152…

```
#spatial plots for wet days per year
fig,ax=plt.subplots(2,5,figsize=(15,5),
                    subplot_kw={'projection': ccrs.PlateCarree()})
ax=ax.flatten()
year_names=['2001','2002','2003','2004','2005','2006','2007','2008','2009','
for i in range(10):
    ax[i].add_feature(cfeature.COASTLINE,linewidth=0.5)
    ax[i].add_feature(cfeature.BORDERS,linewidth=2)
    ax[i].add_feature(cfeature.STATES,linewidth=0.5)
     # ax[i].add_feature(cfeature.OCEAN,linewidth=2)
       # ax[i].add_feature(cfeature.LAKES,linewidth=2 color='blue')
        #    ax[i].add_feature(cfeature.RIVERS)
    ax[i].set_extent([125,152,30,45])

    ax[i].set_title(year_names[i])
    cb=ax[i].contourf(wetDays_per_year_ts.lon,wetDays_per_year_ts.lat,wetDay
    color_bar=fig.add_axes([0.92,0.29,0.025,0.5])
fig.colorbar(cb,cax=color_bar,label='precipitation counts')
plt.subplots_adjust(wspace=-0.10,top=0.93)
plt.suptitle('spatial plots for wet days per year',fontweight='bold')
plt.savefig('wet days per year.jpeg')
```



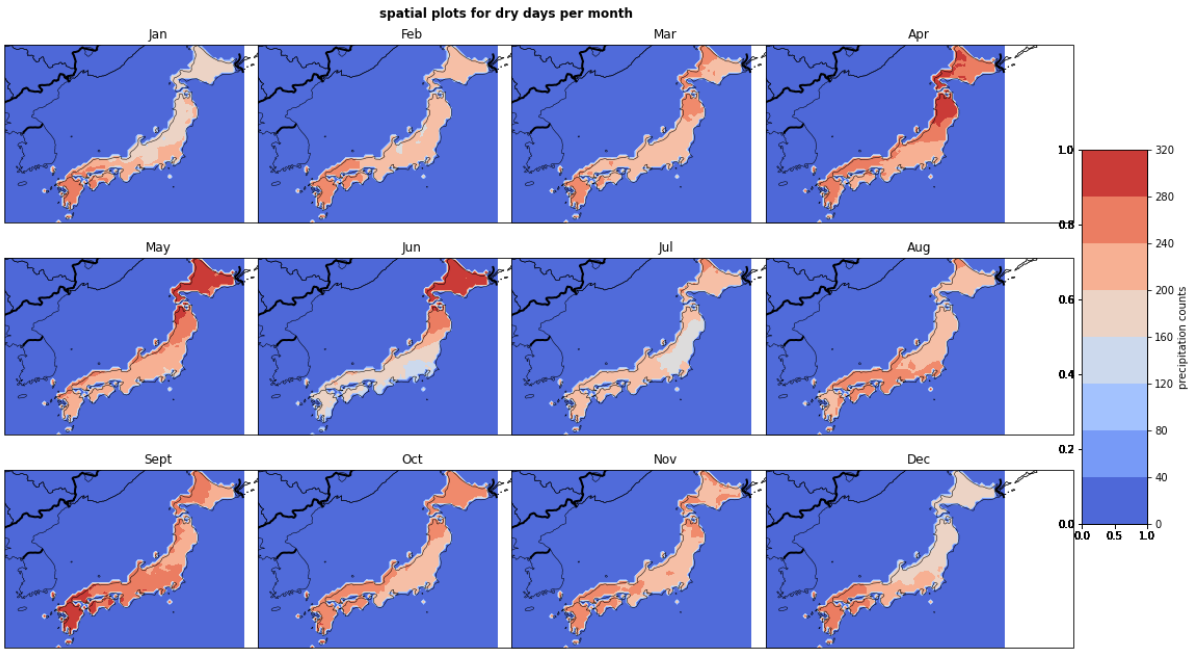
In [144…

```
#spatial plots for wet days per month
fig,ax=plt.subplots(3,4,figsize=(35,10),
                    subplot_kw={'projection': ccrs.PlateCarree()})
ax=ax.flatten()
month_names=['Jan','Feb','Mar','Apr','May','Jun','Jul','Aug','Sept','Oct','N
for i in range(12):
    ax[i].add_feature(cfeature.COASTLINE,linewidth=0.5)
    ax[i].add_feature(cfeature.BORDERS,linewidth=2)
    ax[i].add_feature(cfeature.STATES,linewidth=0.5)
     # ax[i].add_feature(cfeature.OCEAN,linewidth=2)
       # ax[i].add_feature(cfeature.LAKES,linewidth=2 color='blue')
        #    ax[i].add_feature(cfeature.RIVERS)
    ax[i].set_extent([125,152,30,45])

    ax[i].set_title(month_names[i])
    cb=ax[i].contourf(wetDays_per_month_ts.lon,wetDays_per_month_ts.lat,wetD
    color_bar=fig.add_axes([0.72,0.29,0.015,0.5])
fig.colorbar(cb,cax=color_bar,label='precipitation counts')
plt.subplots_adjust(wspace=-0.80,top=0.93)
plt.suptitle('spatial plots for wet days per month',fontweight='bold')
plt.savefig('wet days per month.jpeg')
```



In [147…

```
#spatial plots for dry days per month
fig,ax=plt.subplots(3,4,figsize=(35,10),
                    subplot_kw={'projection': ccrs.PlateCarree()})
ax=ax.flatten()
month_names=['Jan','Feb','Mar','Apr','May','Jun','Jul','Aug','Sept','Oct','N
for i in range(12):
    ax[i].add_feature(cfeature.COASTLINE,linewidth=0.5)
    ax[i].add_feature(cfeature.BORDERS,linewidth=2)
    ax[i].add_feature(cfeature.STATES,linewidth=0.5)
     # ax[i].add_feature(cfeature.OCEAN,linewidth=2)
       # ax[i].add_feature(cfeature.LAKES,linewidth=2 color='blue')
        #    ax[i].add_feature(cfeature.RIVERS)
    ax[i].set_extent([125,152,30,45])

    ax[i].set_title(month_names[i])
    cb=ax[i].contourf(dryDays_per_month_ts.lon,dryDays_per_month_ts.lat,dryD
    color_bar=fig.add_axes([0.72,0.29,0.025,0.5])
fig.colorbar(cb,cax=color_bar,label='precipitation counts')
plt.subplots_adjust(wspace=-0.80,top=0.93)
plt.suptitle('spatial plots for dry days per month',fontweight='bold')
plt.savefig('dry days per month.png')
```



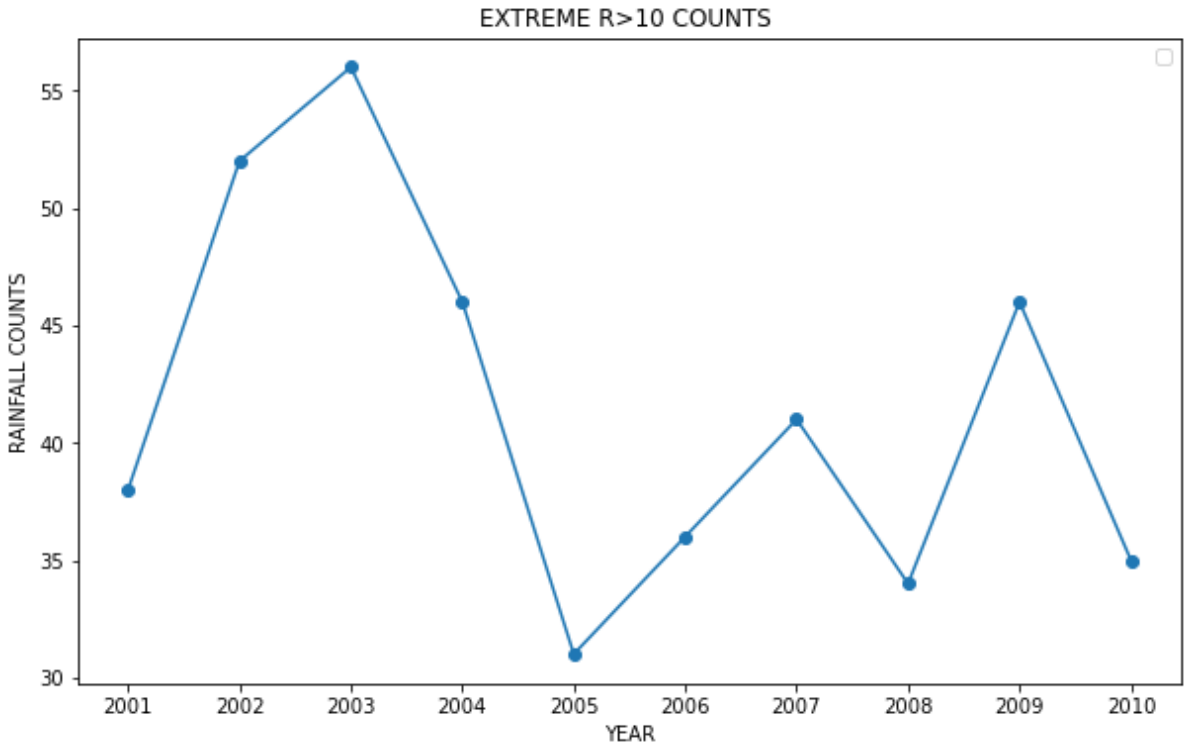
In [159…

```
extreme_10_year_TS = japan_RR.where(japan_RR > 10).groupby('datetime.year').
extreme_20_year_TS = japan_RR.where(japan_RR > 20).groupby('datetime.year').
extreme_10_month_TS = japan_RR.where(japan_RR > 10).groupby('datetime.month'
extreme_20_month_TS = japan_RR.where(japan_RR > 20).groupby('datetime.month'
```

TIME SERIES FOR EXTREME RAINFALL

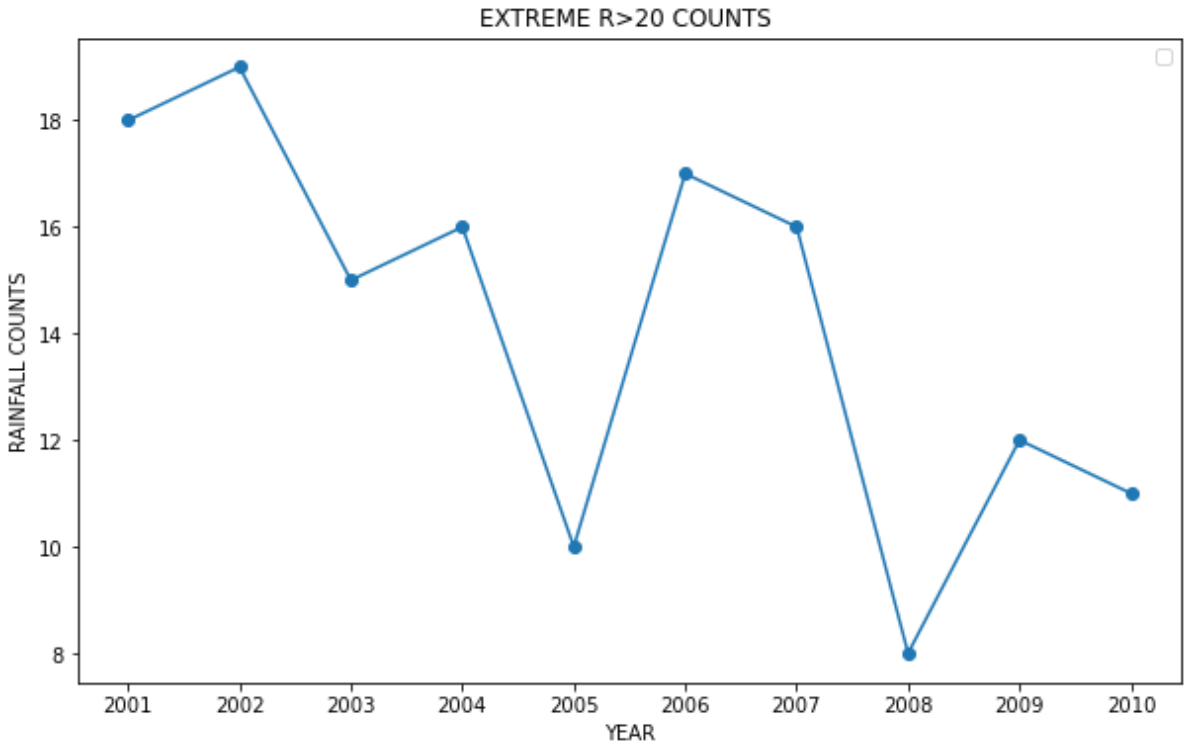
In [160…
```
fig,ax = plt.subplots(figsize=(10,6))
#plt.sublots_adjust(hspace=0.5, wspace=0.2)
plt.plot(year,extreme_10_year_TS, marker='o' )
#plt.plot(year,annual_clime,marker='o', label='precipitation')
plt.xlabel('YEAR')
plt.ylabel('RAINFALL COUNTS')
plt.title('EXTREME R>10 COUNTS ')
plt.legend(loc='upper right')
plt.savefig('timeseries for XTREME10 ')
```

```
No handles with labels found to put in legend.
```



In [165…
```
fig,ax = plt.subplots(figsize=(10,6))
#plt.sublots_adjust(hspace=0.5, wspace=0.2)
plt.plot(year,extreme_20_year_TS, marker='o' )
#plt.plot(year,annual_clime,marker='o', label='precipitation')
plt.xlabel('YEAR')
plt.ylabel('RAINFALL COUNTS')
plt.title('EXTREME R>20 COUNTS ')
plt.legend(loc='upper right')
plt.savefig('timeseries for XTREME20 ')
```

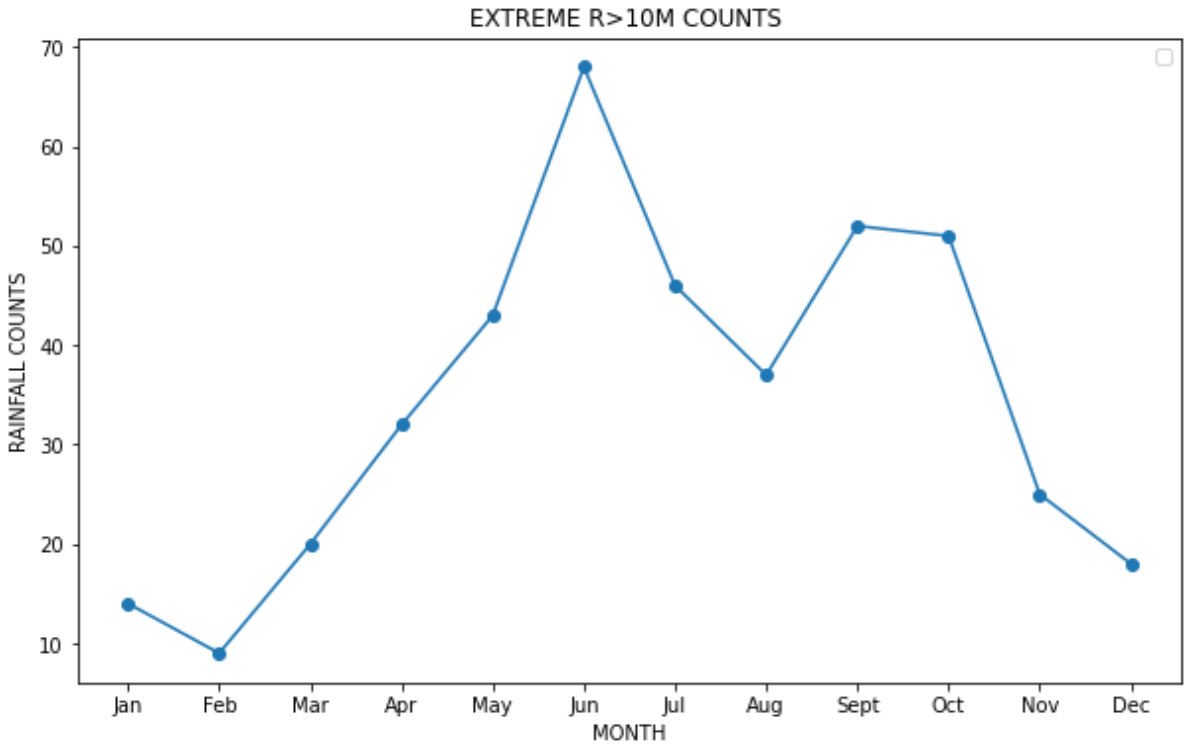
```
No handles with labels found to put in legend.
```



In [166…

```
fig,ax = plt.subplots(figsize=(10,6))
#plt.sublots_adjust(hspace=0.5, wspace=0.2)
plt.plot(month,extreme_10_month_TS, marker='o' )
#plt.plot(year,annual_clime,marker='o', label='precipitation')
plt.xlabel('MONTH')
plt.ylabel('RAINFALL COUNTS')
plt.title('EXTREME R>10M COUNTS ')
plt.legend(loc='upper right')
plt.savefig('timeseries for XTREME10M ')
```

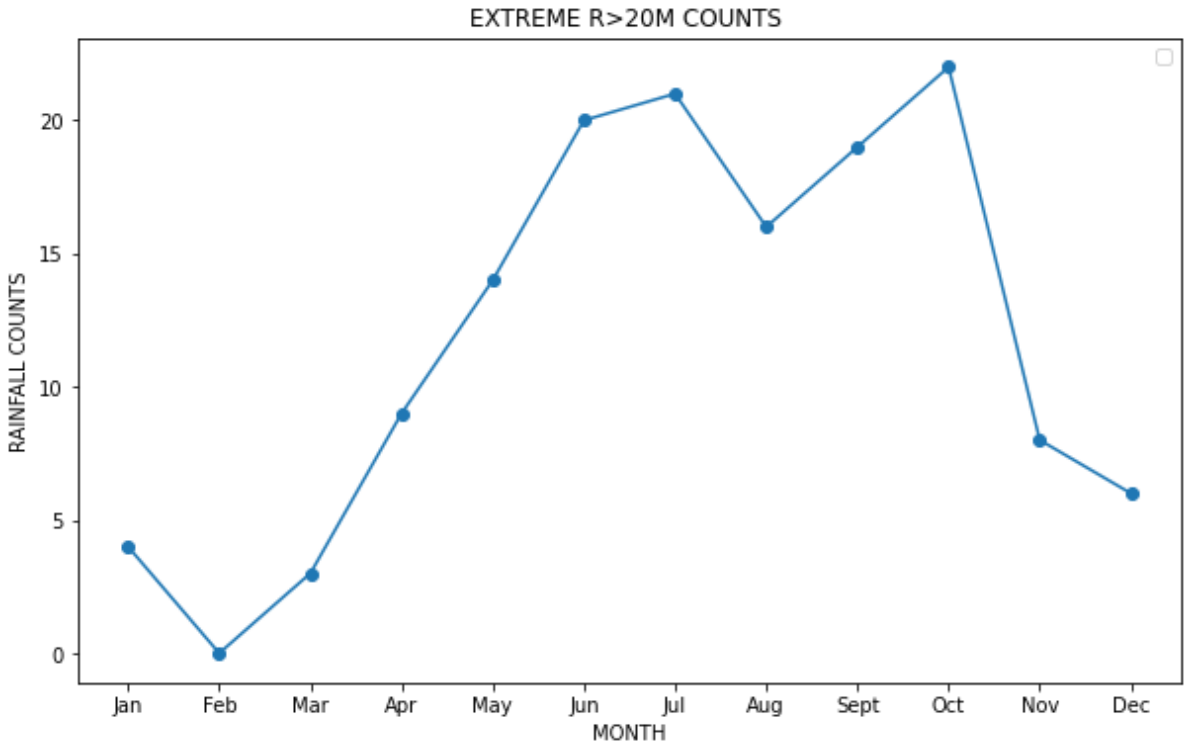
```
No handles with labels found to put in legend.
```



In [167…

```
fig,ax = plt.subplots(figsize=(10,6))
#plt.sublots_adjust(hspace=0.5, wspace=0.2)
plt.plot(month,extreme_20_month_TS, marker='o' )
#plt.plot(year,annual_clime,marker='o', label='precipitation')
plt.xlabel('MONTH')
plt.ylabel('RAINFALL COUNTS')
plt.title('EXTREME R>20M COUNTS ')
plt.legend(loc='upper right')
plt.savefig('timeseries for XTREME20M ')
```

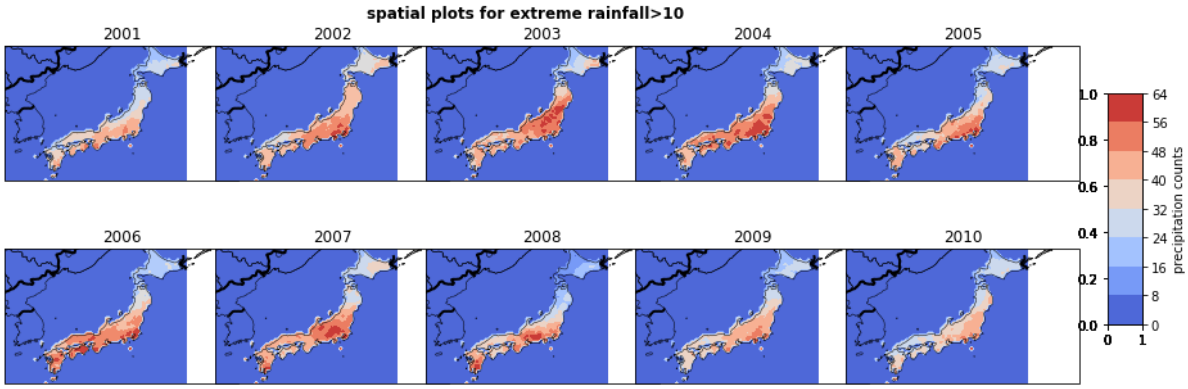
```
No handles with labels found to put in legend.
```



In [162…
```
extreme_10_year_SP = japan_rr.where(japan_rr > 10).groupby('datetime.year').
extreme_20_year_SP = japan_rr.where(japan_rr > 20).groupby('datetime.year').
extreme_10_month_SP = japan_rr.where(japan_rr > 10).groupby('datetime.month'
extreme_20_month_SP = japan_rr.where(japan_rr > 20).groupby('datetime.month'
```

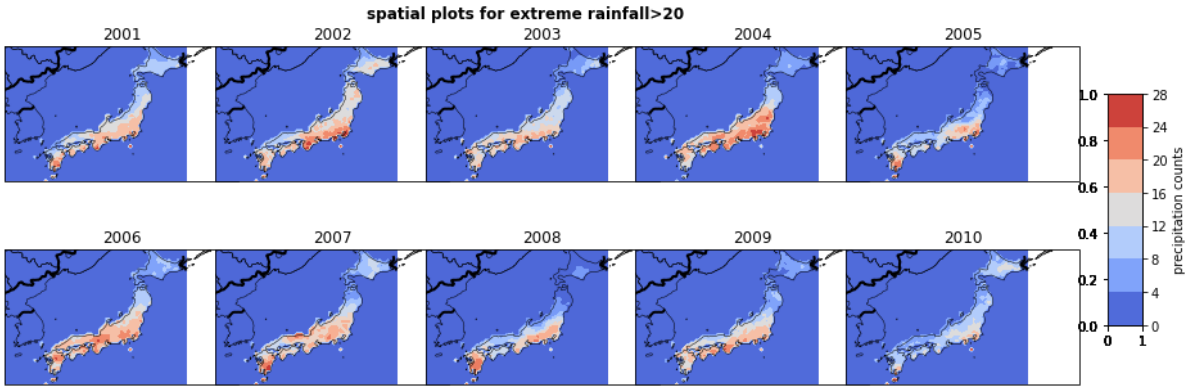
SPATIAL PLOTS FOR EXTREME RAINFALL

In [163…
```
fig,ax=plt.subplots(2,5,figsize=(15,5),
                    subplot_kw={'projection': ccrs.PlateCarree()})
ax=ax.flatten()
year_names=['2001','2002','2003','2004','2005','2006','2007','2008','2009','
for i in range(10):
    ax[i].add_feature(cfeature.COASTLINE,linewidth=0.5)
    ax[i].add_feature(cfeature.BORDERS,linewidth=2)
    ax[i].add_feature(cfeature.STATES,linewidth=0.5)
     # ax[i].add_feature(cfeature.OCEAN,linewidth=2)
       # ax[i].add_feature(cfeature.LAKES,linewidth=2 color='blue')
        #    ax[i].add_feature(cfeature.RIVERS)
    ax[i].set_extent([125,152,30,45])

    ax[i].set_title(year_names[i])
    cb=ax[i].contourf(extreme_10_year_SP.lon,extreme_10_year_SP.lat,extreme_
    color_bar=fig.add_axes([0.92,0.29,0.025,0.5])
fig.colorbar(cb,cax=color_bar,label='precipitation counts')
plt.subplots_adjust(wspace=-0.10,top=0.93)
plt.suptitle('spatial plots for extreme rainfall>10',fontweight='bold')
plt.savefig('extreme10year.png')
```



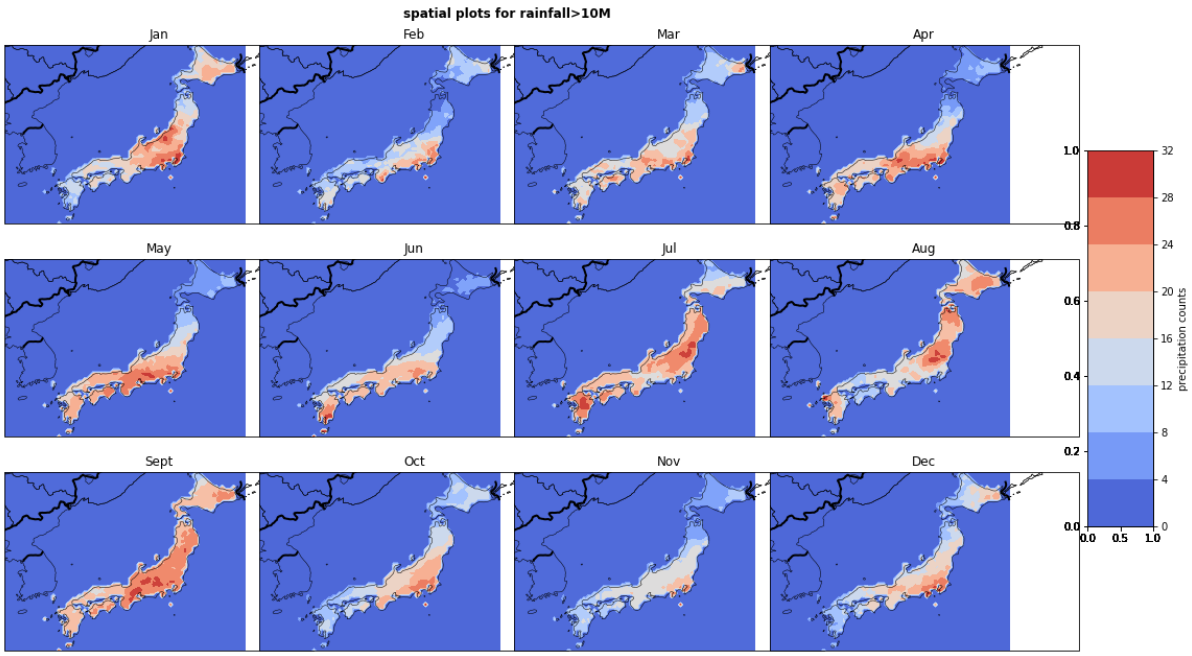
In [164…

```
fig,ax=plt.subplots(2,5,figsize=(15,5),
                    subplot_kw={'projection': ccrs.PlateCarree()})
ax=ax.flatten()
year_names=['2001','2002','2003','2004','2005','2006','2007','2008','2009','
for i in range(10):
    ax[i].add_feature(cfeature.COASTLINE,linewidth=0.5)
    ax[i].add_feature(cfeature.BORDERS,linewidth=2)
    ax[i].add_feature(cfeature.STATES,linewidth=0.5)
     # ax[i].add_feature(cfeature.OCEAN,linewidth=2)
       # ax[i].add_feature(cfeature.LAKES,linewidth=2 color='blue')
        #    ax[i].add_feature(cfeature.RIVERS)
    ax[i].set_extent([125,152,30,45])

    ax[i].set_title(year_names[i])
    cb=ax[i].contourf(extreme_20_year_SP.lon,extreme_20_year_SP.lat,extreme_
    color_bar=fig.add_axes([0.92,0.29,0.025,0.5])
fig.colorbar(cb,cax=color_bar,label='precipitation counts')
plt.subplots_adjust(wspace=-0.10,top=0.93)
plt.suptitle('spatial plots for extreme rainfall>20',fontweight='bold')
plt.savefig('extreme20year.png')
```



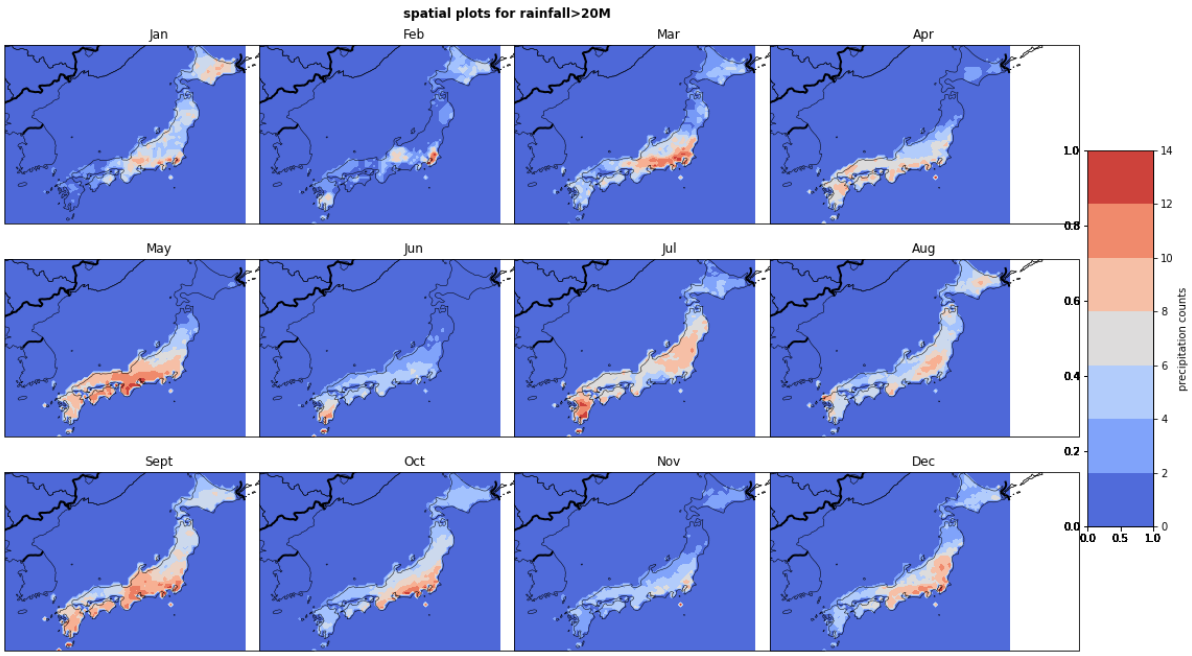
In [168…

```
fig,ax=plt.subplots(3,4,figsize=(35,10),
                    subplot_kw={'projection': ccrs.PlateCarree()})
ax=ax.flatten()
month_names=['Jan','Feb','Mar','Apr','May','Jun','Jul','Aug','Sept','Oct','N
for i in range(12):
    ax[i].add_feature(cfeature.COASTLINE,linewidth=0.5)
    ax[i].add_feature(cfeature.BORDERS,linewidth=2)
    ax[i].add_feature(cfeature.STATES,linewidth=0.5)
     # ax[i].add_feature(cfeature.OCEAN,linewidth=2)
       # ax[i].add_feature(cfeature.LAKES,linewidth=2 color='blue')
        #    ax[i].add_feature(cfeature.RIVERS)
    ax[i].set_extent([125,152,30,45])

    ax[i].set_title(month_names[i])
    cb=ax[i].contourf(extreme_10_month_SP.lon,extreme_10_month_SP.lat,extrem
    color_bar=fig.add_axes([0.72,0.29,0.025,0.5])
fig.colorbar(cb,cax=color_bar,label='precipitation counts')
plt.subplots_adjust(wspace=-0.80,top=0.93)
plt.suptitle('spatial plots for rainfall>10M ',fontweight='bold')
plt.savefig('extremerainfall10month.png')
```



In [169…

```
fig,ax=plt.subplots(3,4,figsize=(35,10),
                    subplot_kw={'projection': ccrs.PlateCarree()})
ax=ax.flatten()
month_names=['Jan','Feb','Mar','Apr','May','Jun','Jul','Aug','Sept','Oct','N
for i in range(12):
    ax[i].add_feature(cfeature.COASTLINE,linewidth=0.5)
    ax[i].add_feature(cfeature.BORDERS,linewidth=2)
    ax[i].add_feature(cfeature.STATES,linewidth=0.5)
     # ax[i].add_feature(cfeature.OCEAN,linewidth=2)
       # ax[i].add_feature(cfeature.LAKES,linewidth=2 color='blue')
        #    ax[i].add_feature(cfeature.RIVERS)
    ax[i].set_extent([125,152,30,45])

    ax[i].set_title(month_names[i])
    cb=ax[i].contourf(extreme_20_month_SP.lon,extreme_20_month_SP.lat,extrem
    color_bar=fig.add_axes([0.72,0.29,0.025,0.5])
fig.colorbar(cb,cax=color_bar,label='precipitation counts')
plt.subplots_adjust(wspace=-0.80,top=0.93)
plt.suptitle('spatial plots for rainfall>20M ',fontweight='bold')
plt.savefig('extremerainfall20month.png')
```



THE END

In [ ]: