Poisson_MGWR_MonteCarlo_Results

May 19, 2020

Notebook Outline:

Monte Carlo experiment code can be found in path mgwr/notebooks/Poisson_MC_script/

0.0.1 Set up Cell

```
In [1]: import warnings
        warnings.filterwarnings("ignore")
        import pickle
        import sys
        import seaborn as sns
        import numpy as np
        sys.path.append("C:/Users/msachde1/Downloads/Research/Development/mgwr/notebooks/Poiss
        import f_2
        import matplotlib.pyplot as plt
        import pandas as pd
        from mpl_toolkits.axes_grid1 import make_axes_locatable
```

```
C:\Users\msachde1\AppData\Local\Continuum\anaconda3\envs\gwrenv\lib\site-packages\libpysal\io\
  warnings.warn('SQLAlchemy and Geomet not installed, database I/O disabled')
```

0.0.2 List bandwidths from pickles

```
In [2]: mgwr_bw0=[]
        mgwr_bw1=[]
        mgwr_bw2=[]
        gwr_bw=[]
```

```
In [3]: for i in range(0,1000,50):
            p1 = pickle.load( open( "C:/Users/msachde1/Downloads/Research/Development/mgwr/note
            for j in range(50):
                mgwr_bw0.append(p1[j].mgwr_bw[0][0])
                mgwr_bw1.append(p1[j].mgwr_bw[0][1])
                mgwr_bw2.append(p1[j].mgwr_bw[0][2])
                gwr_bw.append(p1[j].gwr_bw[0])
```

0.0.3 Parameter functions

```
In [4]: def add(a,b):
            return 1+((1/120)*(a+b))

        def con(u,v):
            return (0*(u)*(v))+0.3

        def sp(u,v):
            return 1+1/3240*(36-(6-u/2)**2)*(36-(6-v/2)**2)

        def med(u,v):
            B = np.zeros((25,25))
            for i in range(25):
                for j in range(25):

                    if u[i][j]<=8:
                        B[i][j]=0.2
                    elif u[i][j]>17:
                        B[i][j]=0.7
                    else:
                        B[i][j]=0.5
            return B
```

```
In [5]: x = np.linspace(0, 25, 25)
        y = np.linspace(25, 0, 25)
        X, Y = np.meshgrid(x, y)

        B0=con(X,Y)
        #B1=add(X,Y)
        B1=sp(X,Y)
        B2=med(X,Y)
```

```
In [104]: x = np.linspace(0, 25, 25)
          y = np.linspace(25, 0, 25)
```

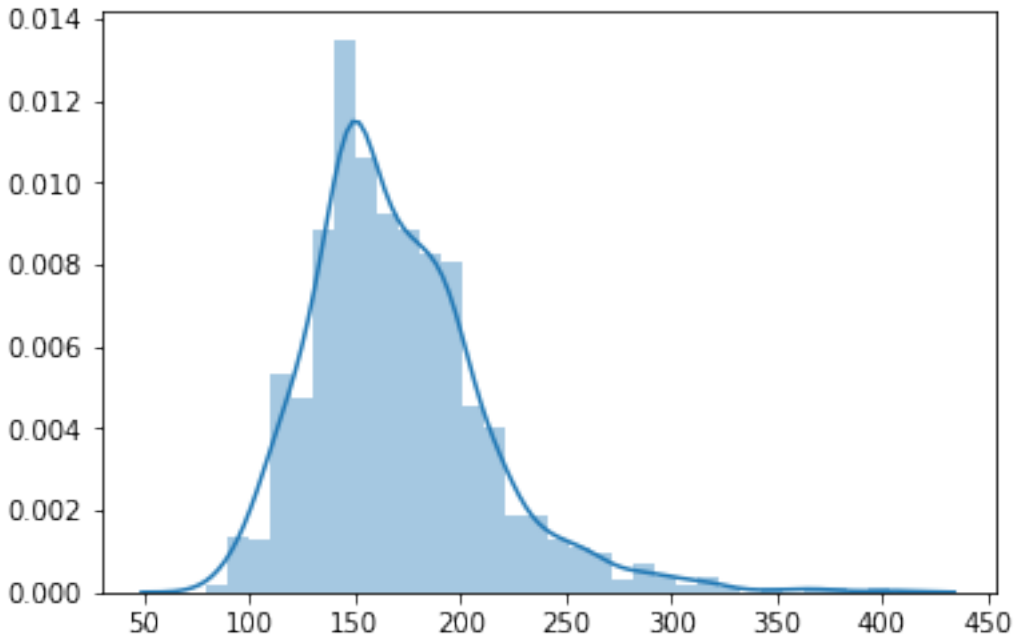
```
In [107]: x = np.linspace(0, 25, 25)
          y = np.linspace(25, 0, 25)
```

```
In [108]: x
```

```
Out[108]: array([ 0.        ,  1.04166667,  2.08333333,  3.125     ,  4.16666667,
                  5.20833333,  6.25      ,  7.29166667,  8.33333333,  9.375     ,
                 10.41666667, 11.45833333, 12.5       , 13.54166667, 14.58333333,
                 15.625     , 16.66666667, 17.70833333, 18.75      , 19.79166667,
                 20.83333333, 21.875     , 22.91666667, 23.95833333, 25.        ])

In [106]: x

Out[106]: array([ 0.        ,  1.04166667,  2.08333333,  3.125     ,  4.16666667,
                  5.20833333,  6.25      ,  7.29166667,  8.33333333,  9.375     ,
                 10.41666667, 11.45833333, 12.5       , 13.54166667, 14.58333333,
                 15.625     , 16.66666667, 17.70833333, 18.75      , 19.79166667,
                 20.83333333, 21.875     , 22.91666667, 23.95833333, 25.        ])
```

0.0.4 GWR bandwidth

```
In [6]: sns.distplot(gwr_bw)

Out[6]: <matplotlib.axes._subplots.AxesSubplot at 0x12a79841da0>
```

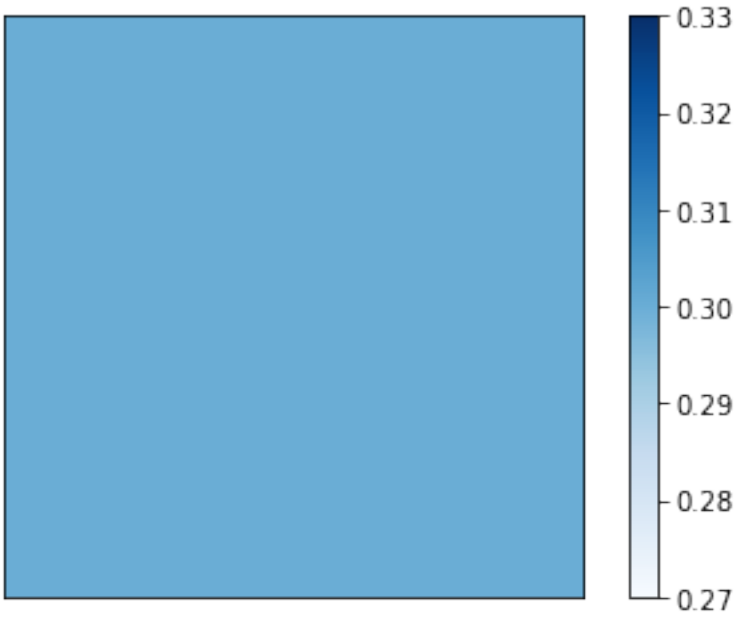



```
In [7]: np.mean(gwr_bw)

Out[7]: 170.074
```

0.0.5 MGWR bandwidths

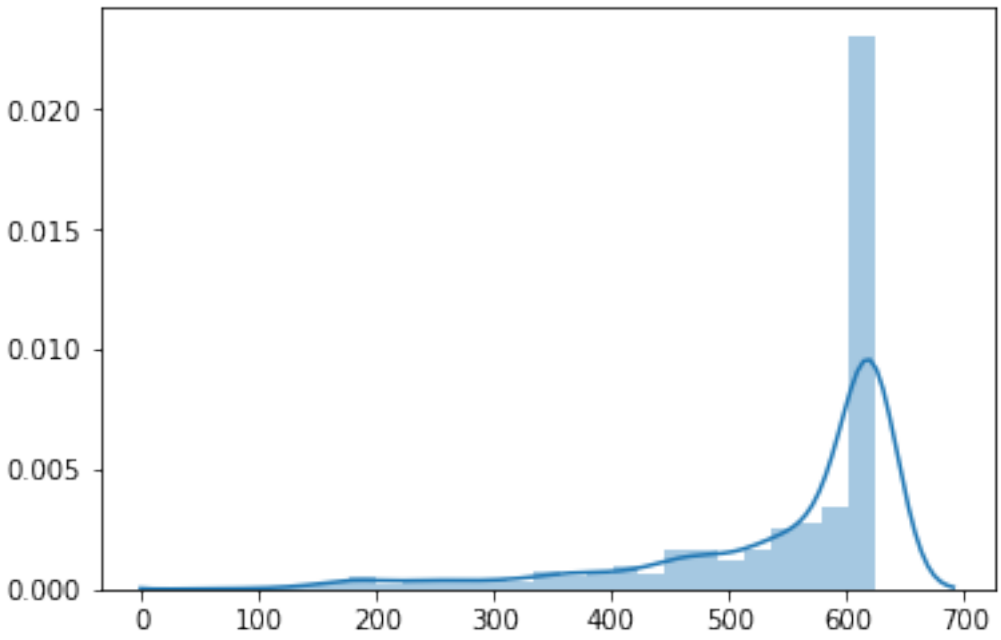
In [8]:
```
plt.imshow(B0, extent=[0,10, 0, 10], origin='lower',cmap='Blues')
plt.colorbar()
plt.axis(aspect='image')
plt.xticks([])
plt.yticks([])
```

Out[8]: ([], <a list of 0 Text yticklabel objects>)



In [9]:
```
sns.distplot(mgwr_bw0)
```

Out[9]: <matplotlib.axes._subplots.AxesSubplot at 0x12a79ca4390>

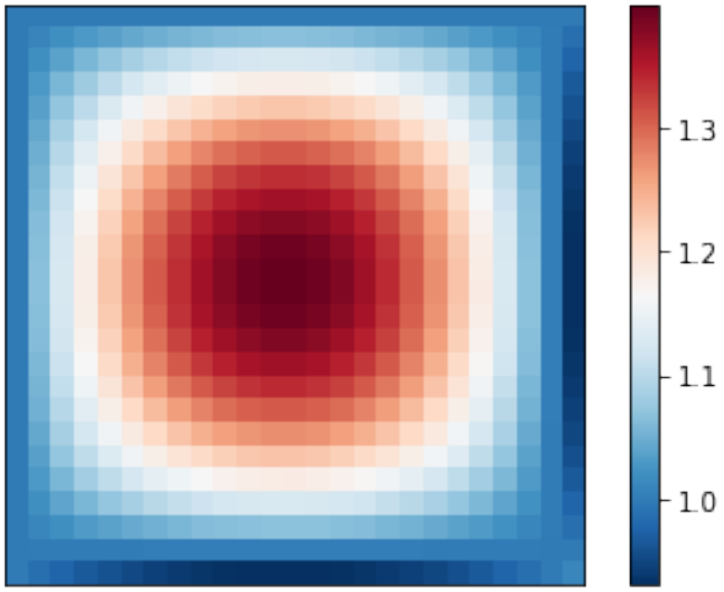



```
In [10]: np.mean(mgwr_bw0)
```

```
Out[10]: 546.083
```

```
In [11]: plt.imshow(B1, extent=[0,25, 0, 25], origin='lower',cmap='RdBu_r')
         plt.colorbar()
         plt.axis(aspect='image')
         plt.xticks([])
         plt.yticks([])
```

```
Out[11]: ([], <a list of 0 Text yticklabel objects>)
```

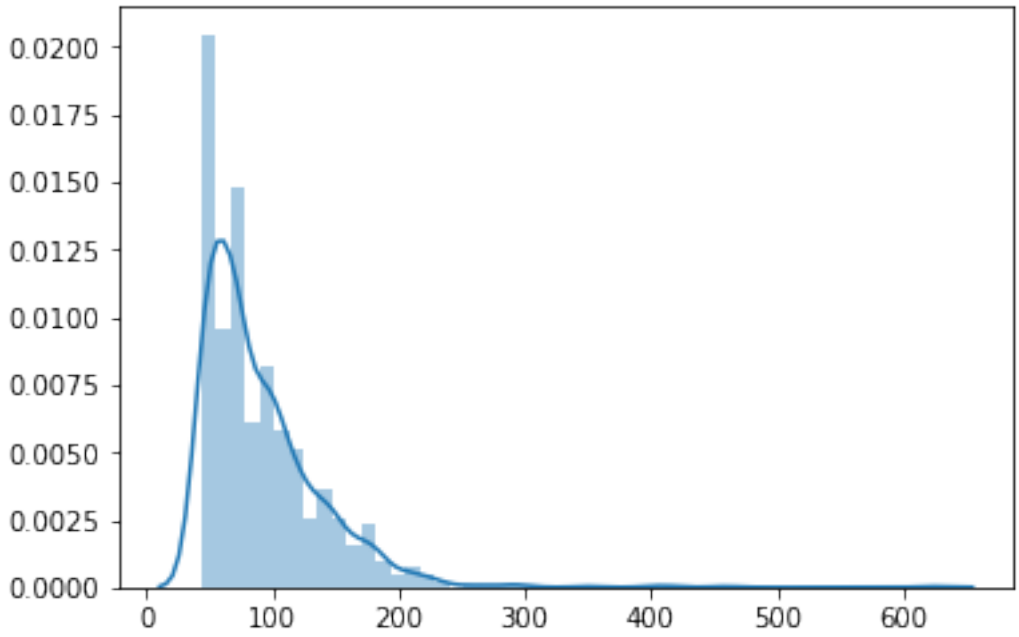



```
In [12]: np.mean(mgwr_bw1)
```

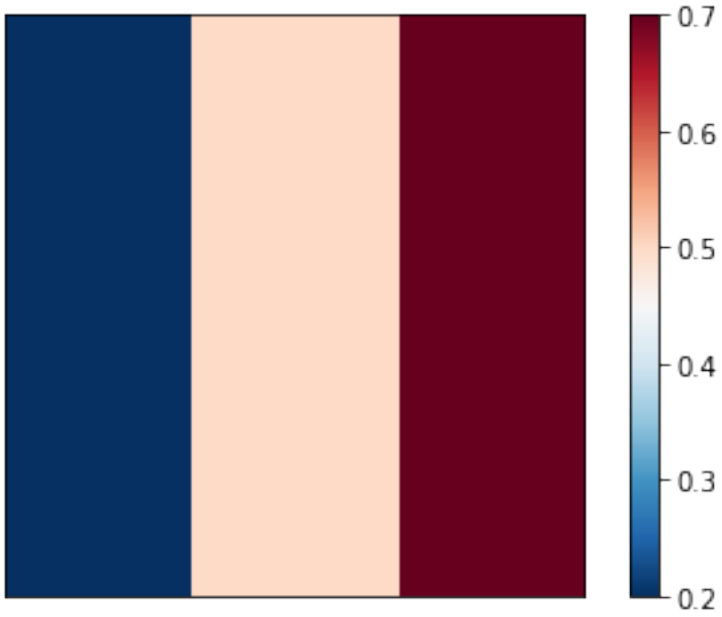
```
Out[12]: 91.753
```

```
In [13]: sns.distplot(mgwr_bw1)
```

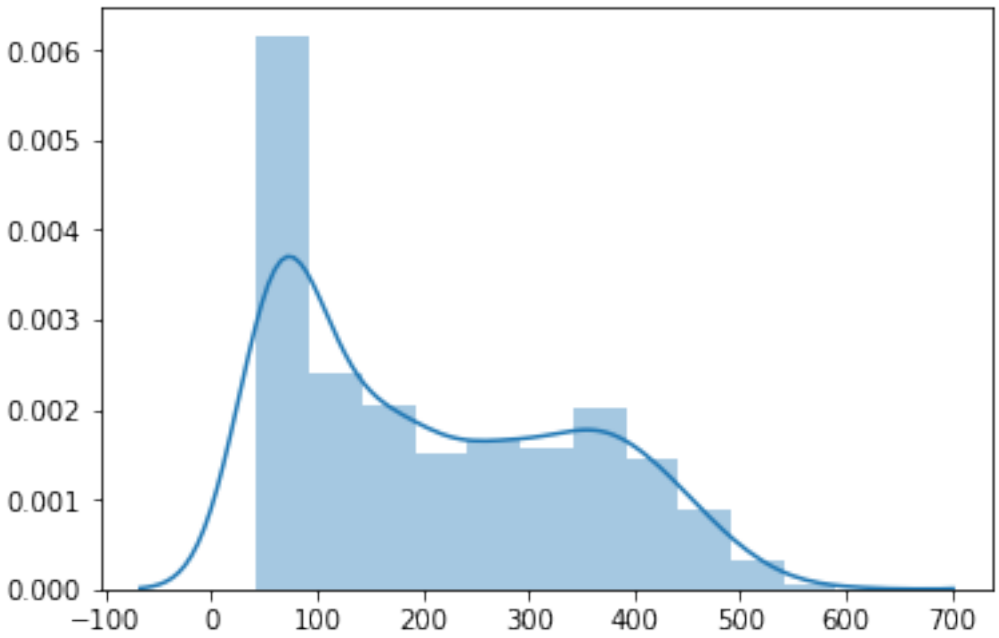
```
Out[13]: <matplotlib.axes._subplots.AxesSubplot at 0x12a79d10e80>
```



```
In [14]: plt.imshow(B2, extent=[0,25, 0, 25], origin='lower',cmap='RdBu_r')
         plt.colorbar()
         plt.axis(aspect='image')
         plt.xticks([])
         plt.yticks([])

Out[14]: ([], <a list of 0 Text yticklabel objects>)
```




```
In [15]: np.mean(mgwr_bw2)

Out[15]: 209.398

In [16]: sns.distplot(mgwr_bw2)

Out[16]: <matplotlib.axes._subplots.AxesSubplot at 0x12a7ae39128>
```




```
In [17]: np.mean(mgwr_bw0),np.mean(mgwr_bw1),np.mean(mgwr_bw2)
```

```
Out[17]: (546.083, 91.753, 209.398)
```

0.0.6 AIC, AICc, BIC check

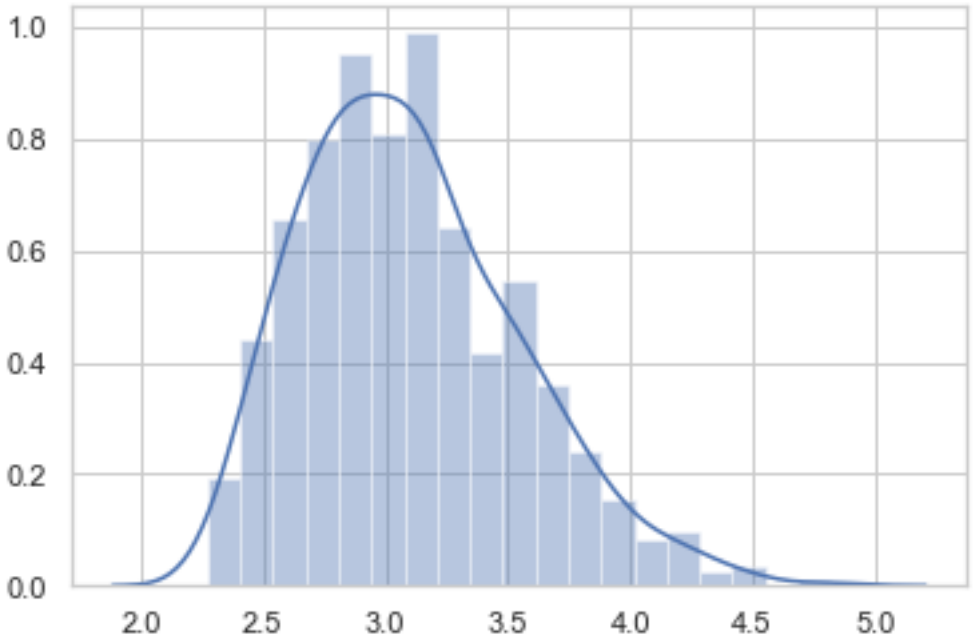
```
In [145]: mgwr_aicc=[]
          gwr_aicc=[]
          mgwr_bic=[]
          gwr_bic=[]
          mgwr_aic=[]
          gwr_aic=[]
          mgwr_params=[]
          gwr_params=[]
          mgwr_predy=[]
          gwr_predy=[]
```

```
In [146]: for i in range(0,1000,50):
              p1 = pickle.load( open( "C:/Users/msachde1/Downloads/Research/Development/mgwr/n
              for j in range(50):
                  mgwr_aicc.append(p1[j].mgwr_aicc[0])
                  gwr_aicc.append(p1[j].gwr_aicc[0])

                  mgwr_bic.append(p1[j].mgwr_bic[0])
                  gwr_bic.append(p1[j].gwr_bic[0])
```

```
                mgwr_aic.append(p1[j].mgwr_aic[0])
                gwr_aic.append(p1[j].gwr_aic[0])

                mgwr_params.append(p1[j].mgwr_params[0])
                gwr_params.append(p1[j].gwr_params[0])

                mgwr_predy.append(p1[j].mgwr_predy[0])
                gwr_predy.append(p1[j].gwr_predy[0])
```

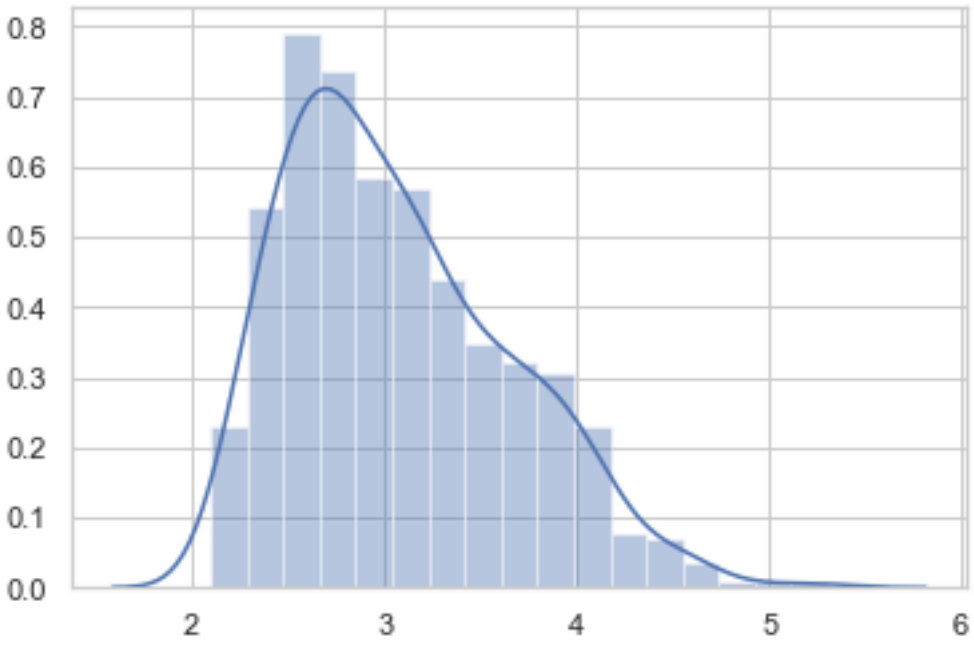
In [148]:
```
sns.distplot(np.mean(gwr_predy,axis=0))
```

Out[148]:
```
<matplotlib.axes._subplots.AxesSubplot at 0x12a79cf1240>
```



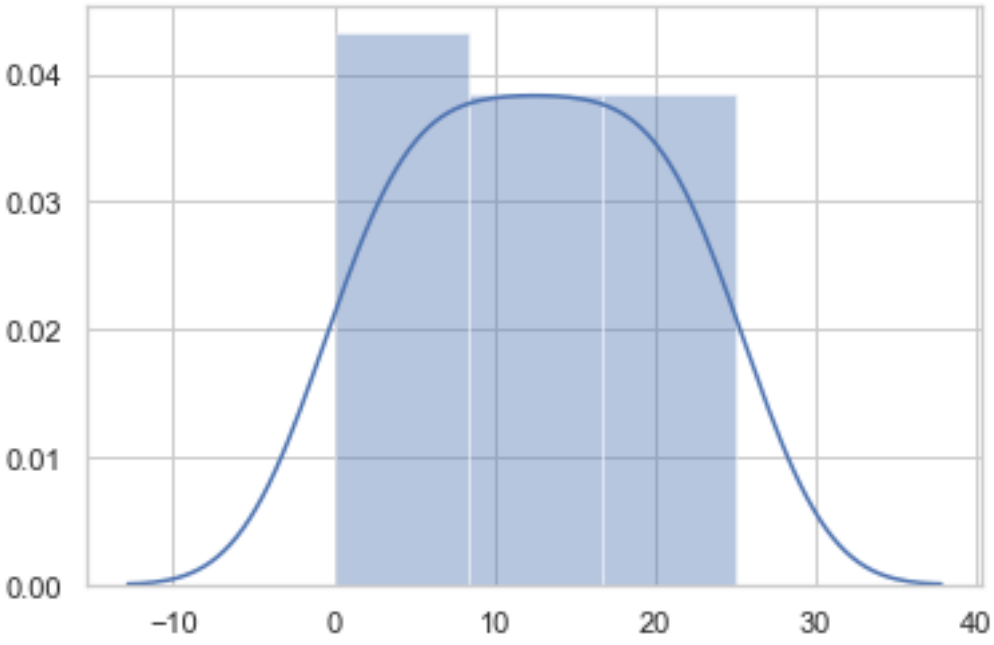
In [150]:
```
sns.distplot(np.mean(mgwr_predy,axis=0))
```

Out[150]:
```
<matplotlib.axes._subplots.AxesSubplot at 0x12a79c3cdd8>
```

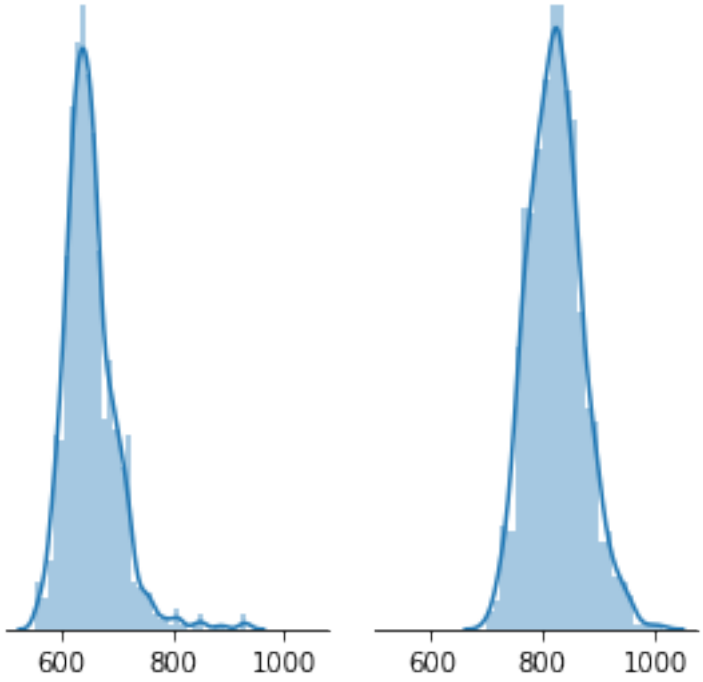



```
In [152]: sns.distplot(y)
```

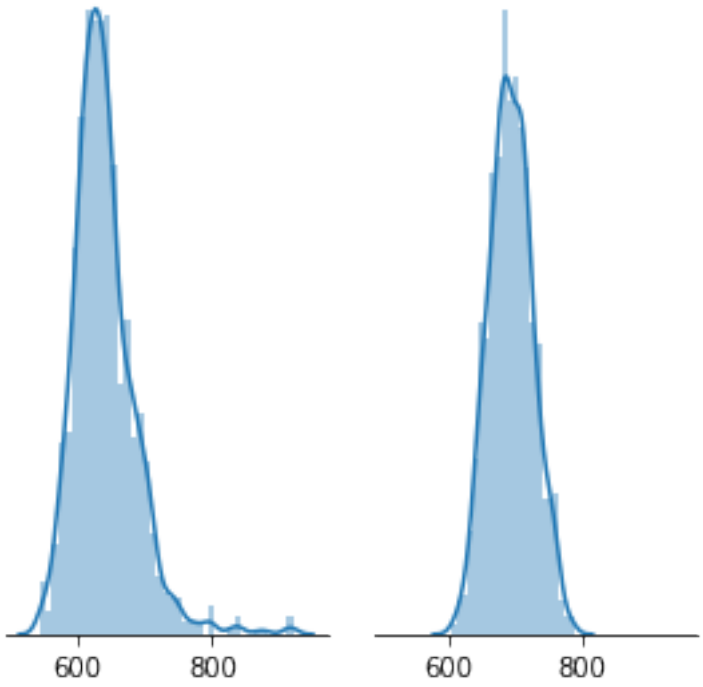
```
Out[152]: <matplotlib.axes._subplots.AxesSubplot at 0x12a7ae935c0>
```



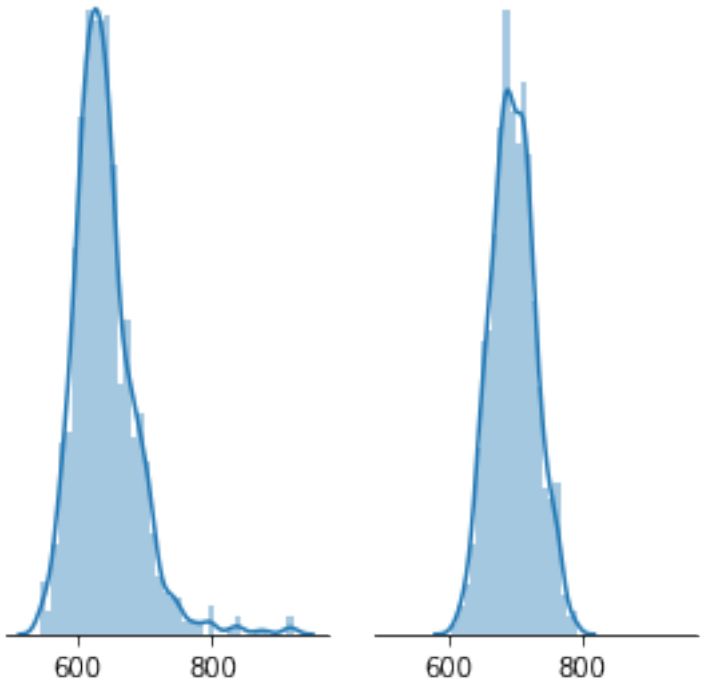
In [20]:
```
f, axes = plt.subplots(1, 2, figsize=(4, 4), sharex=True)
sns.despine(left=True)
sns.distplot(mgwr_bic,ax=axes[0])
sns.distplot(gwr_bic,ax=axes[1])

plt.setp(axes, yticks=[])
plt.tight_layout()
```



In [21]:
```
f, axes = plt.subplots(1, 2, figsize=(4, 4), sharex=True)
sns.despine(left=True)
sns.distplot(mgwr_aic,ax=axes[0])
sns.distplot(gwr_aic,ax=axes[1])

plt.setp(axes, yticks=[])
plt.tight_layout()
```




```
In [22]: f, axes = plt.subplots(1, 2, figsize=(4, 4), sharex=True)
         sns.despine(left=True)
         sns.distplot(mgwr_aicc,ax=axes[0])
         sns.distplot(gwr_aicc,ax=axes[1])

         plt.setp(axes, yticks=[])
         plt.tight_layout()
```




```
In [23]: np.mean(mgwr_aicc), np.mean(gwr_aicc)
```

```
Out[23]: (640.2736332530651, 696.9264487767485)
```

```
In [24]: np.mean(mgwr_aic), np.mean(gwr_aic)
```

```
Out[24]: (640.2353688823071, 693.79977380067)
```

```
In [25]: np.mean(mgwr_bic), np.mean(gwr_bic)
```

```
Out[25]: (653.4726886933404, 823.1916659552306)
```

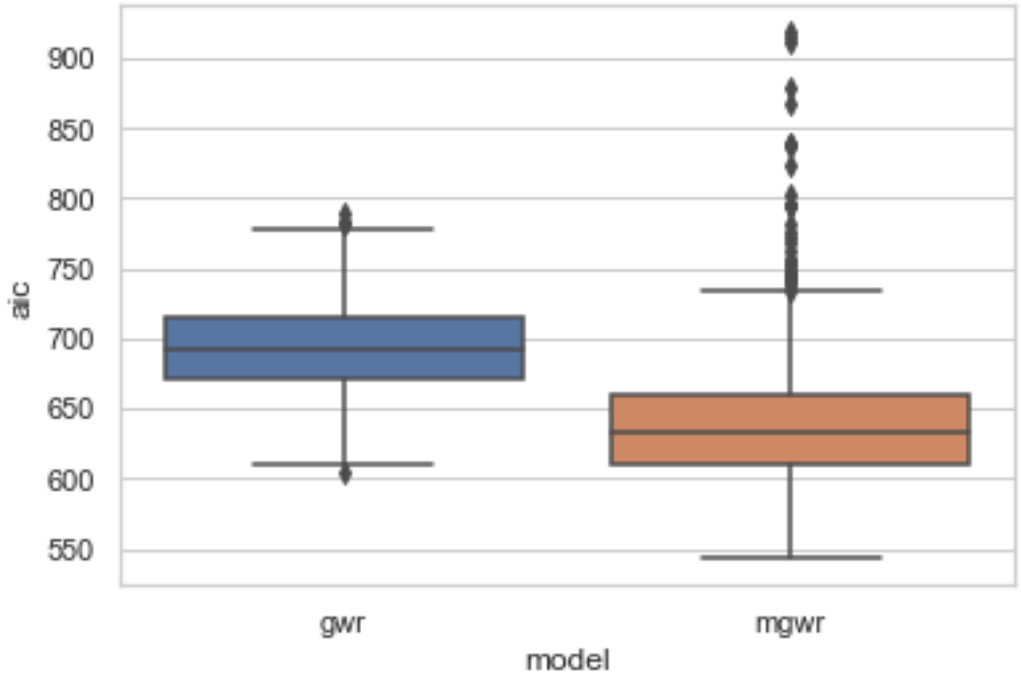
AIC, AICc, BIC Boxplots for comparison

```
In [26]: model=[]
         model = ['gwr']*1000
         model2 = ['mgwr']*1000
         model=model+model2
```

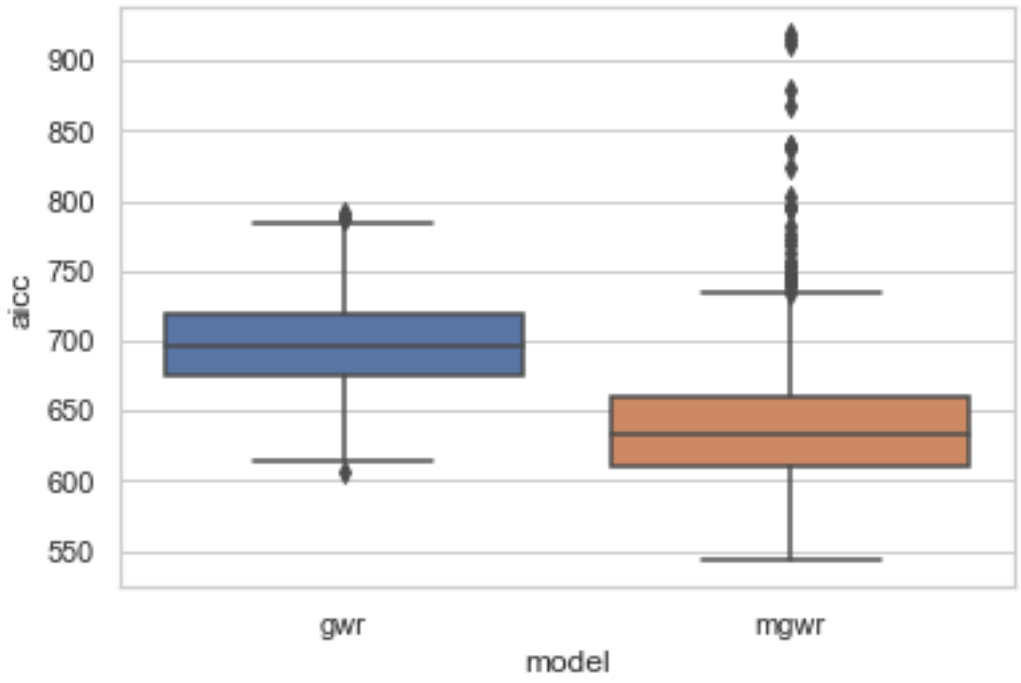
```
In [27]: aic=[]
         aic=gwr_aic
         aic=aic+mgwr_aic
```

```
In [28]: aicc=[]
         aicc=gwr_aicc
         aicc=aicc+mgwr_aicc
```

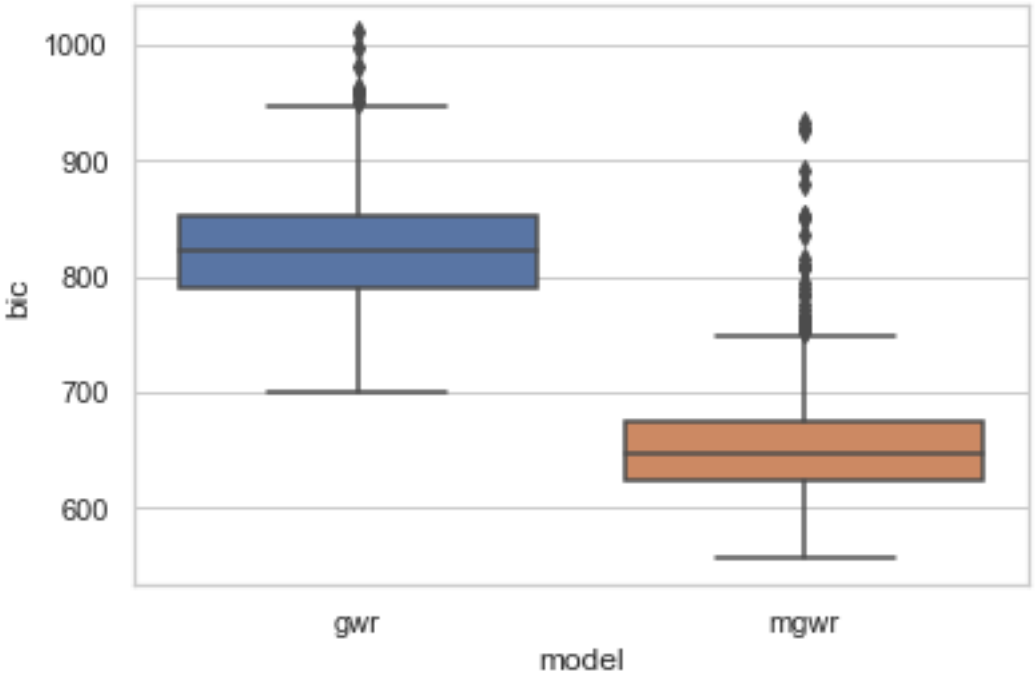
```
In [29]: bic=[]
         bic=gwr_bic
         bic=bic+mgwr_bic

In [30]: d = {'aic':aic,'bic':bic,'aicc':aicc,'model':model}

In [31]: df=pd.DataFrame(data=d)

In [32]: sns.set(style="whitegrid")
         ax = sns.boxplot(y=df['aic'],x=df['model'])
```




```
In [33]: sns.set(style="whitegrid")
         ax = sns.boxplot(y=df['aicc'],x=df['model'])
```



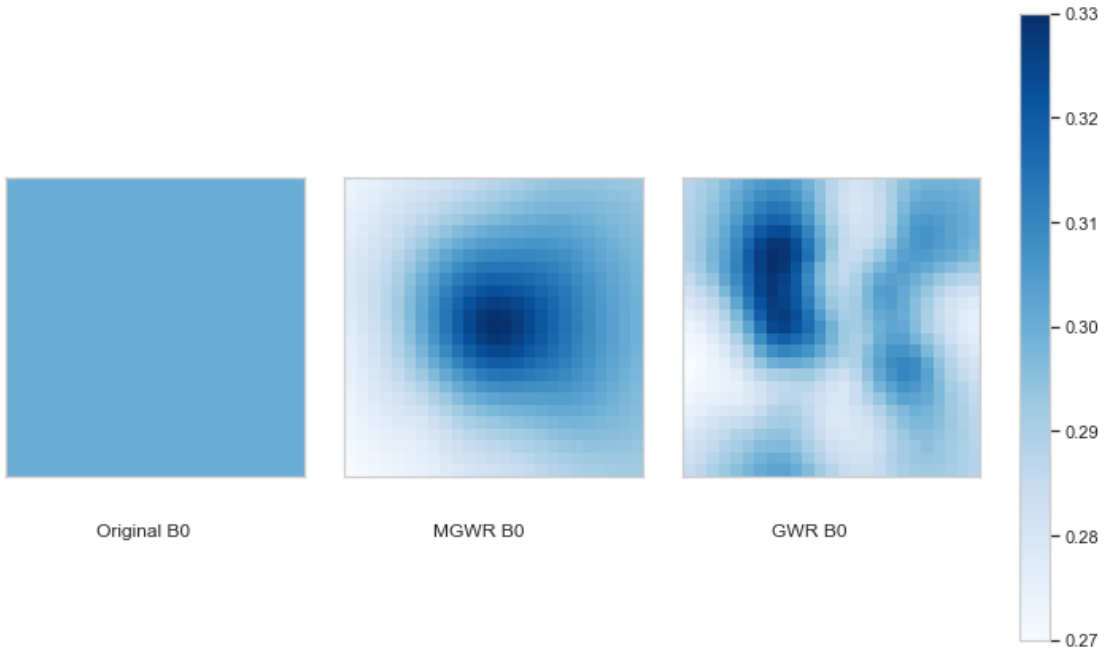

```
In [34]: sns.set(style="whitegrid")
         ax = sns.boxplot(y=df['bic'],x=df['model'])
```



0.0.7 Parameter comparison from MGWR and GWR

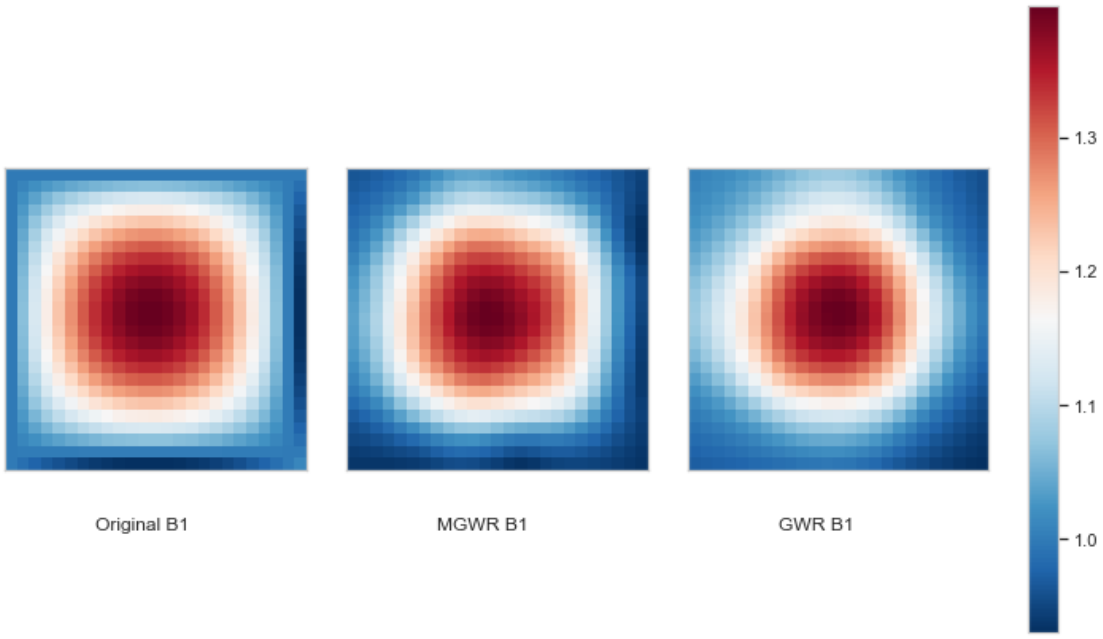
```
In [35]: mgwr_params_mean=np.mean(mgwr_params,axis=0)
         gwr_params_mean=np.mean(gwr_params,axis=0)

In [36]: gwr_params_mean

Out[36]: array([[0.29918256, 1.08091908, 0.24303902],
                [0.29986681, 1.08191172, 0.25072761],
                [0.30086733, 1.08323714, 0.26047537],
                ...,
                [0.30379225, 1.08046329, 0.66564584],
                [0.30356667, 1.07606719, 0.67132769],
                [0.30334916, 1.07243615, 0.67578556]])

In [37]: B0_mgwr=np.hsplit(mgwr_params_mean,3)[0]
         B1_mgwr=np.hsplit(mgwr_params_mean,3)[1]
         B2_mgwr=np.hsplit(mgwr_params_mean,3)[2]

In [38]: B0_gwr=np.hsplit(gwr_params_mean,3)[0]
         B1_gwr=np.hsplit(gwr_params_mean,3)[1]
         B2_gwr=np.hsplit(gwr_params_mean,3)[2]

In [39]: B0_mgwr=B0_mgwr.reshape(25,25)
         B1_mgwr=B1_mgwr.reshape(25,25)
         B2_mgwr=B2_mgwr.reshape(25,25)

In [40]: B0_gwr=B0_gwr.reshape(25,25)
         B1_gwr=B1_gwr.reshape(25,25)
         B2_gwr=B2_gwr.reshape(25,25)

In [41]: fig, (ax, ax2,ax3, cax) = plt.subplots(ncols=4,figsize=(10,6),
                                 gridspec_kw={"width_ratios":[1,1,1, 0.1],"height_ratios":[1]})
         fig.subplots_adjust(wspace=0.3)
         im = ax.imshow(B0, extent=[0,10, 0, 10], origin='lower',cmap='Blues')
         ax.text(3, -2, 'Original B0')
         im2  = ax2.imshow(B0_mgwr, extent=[0,10, 0, 10], origin='lower',cmap='Blues')
         ax2.text(3, -2, 'MGWR B0')
         im3 = ax3.imshow(B0_gwr, extent=[0,10, 0, 10], origin='lower',cmap='Blues')
         ax3.text(3, -2, 'GWR B0')

         divider = make_axes_locatable(ax3)

         fig.colorbar(im, cax=cax)

         ax.set_xticks([])
```

```
ax.set_yticks([])

ax2.set_xticks([])
ax2.set_yticks([])

ax3.set_xticks([])
ax3.set_yticks([])

plt.tight_layout()
```



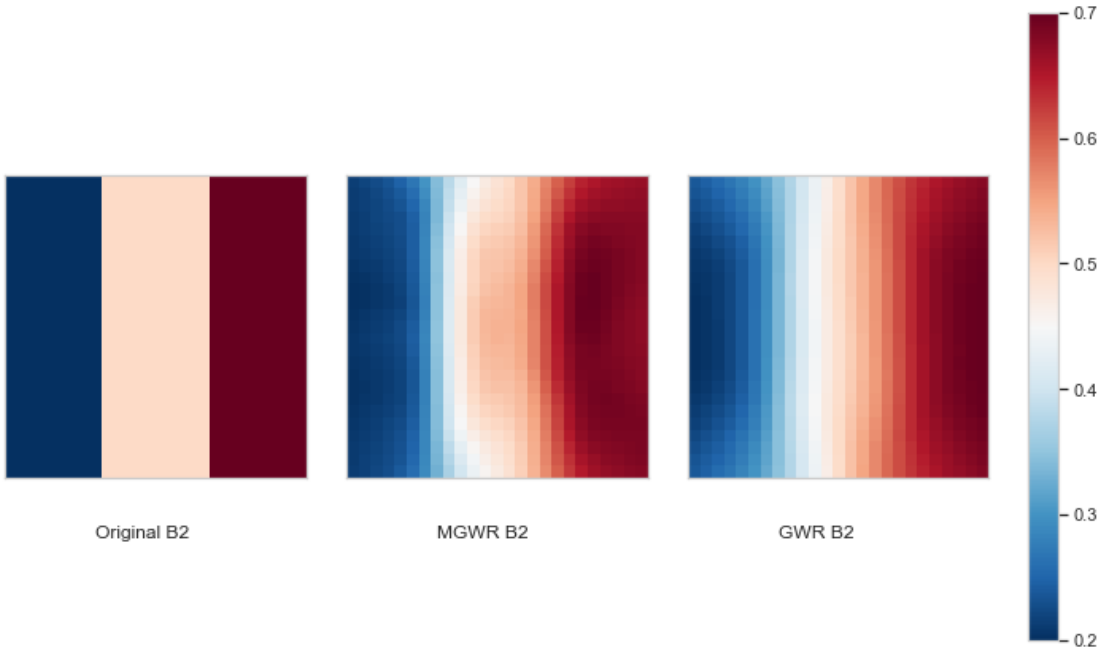
In [42]:
```
fig, (ax, ax2,ax3, cax) = plt.subplots(ncols=4,figsize=(10,6),
                         gridspec_kw={"width_ratios":[1,1,1, 0.1],"height_ratios":[1]})
fig.subplots_adjust(wspace=0.3)
im = ax.imshow(B1, extent=[0,10, 0, 10], origin='lower',cmap='RdBu_r')
ax.text(3, -2, 'Original B1')
im2  = ax2.imshow(B1_mgwr, extent=[0,10, 0, 10], origin='lower',cmap='RdBu_r')
ax2.text(3, -2, 'MGWR B1')
im3 = ax3.imshow(B1_gwr, extent=[0,10, 0, 10], origin='lower',cmap='RdBu_r')
ax3.text(3, -2, 'GWR B1')

divider = make_axes_locatable(ax3)

fig.colorbar(im, cax=cax)

ax.set_xticks([])
ax.set_yticks([])
```

```
ax2.set_xticks([])
ax2.set_yticks([])

ax3.set_xticks([])
ax3.set_yticks([])

plt.tight_layout()
```



In [43]:
```
fig, (ax, ax2,ax3, cax) = plt.subplots(ncols=4,figsize=(10,6),
                        gridspec_kw={"width_ratios":[1,1,1, 0.1],"height_ratios":[1]})
fig.subplots_adjust(wspace=0.3)
im = ax.imshow(B2, extent=[0,10, 0, 10], origin='lower',cmap='RdBu_r')
ax.text(3, -2, 'Original B2')
im2  = ax2.imshow(B2_mgwr, extent=[0,10, 0, 10], origin='lower',cmap='RdBu_r')
ax2.text(3, -2, 'MGWR B2')
im3 = ax3.imshow(B2_gwr, extent=[0,10, 0, 10], origin='lower',cmap='RdBu_r')
ax3.text(3, -2, 'GWR B2')

divider = make_axes_locatable(ax3)

fig.colorbar(im, cax=cax)

ax.set_xticks([])
ax.set_yticks([])
```

```
ax2.set_xticks([])
ax2.set_yticks([])

ax3.set_xticks([])
ax3.set_yticks([])

plt.tight_layout()
```



0.0.8 Comparing parameters (MGWR and GWR)

$$RMSE_j = \sqrt{1/n \sum (\beta_j(u_i, v_i) - \hat{\beta}(u_i, v_i))^2}$$

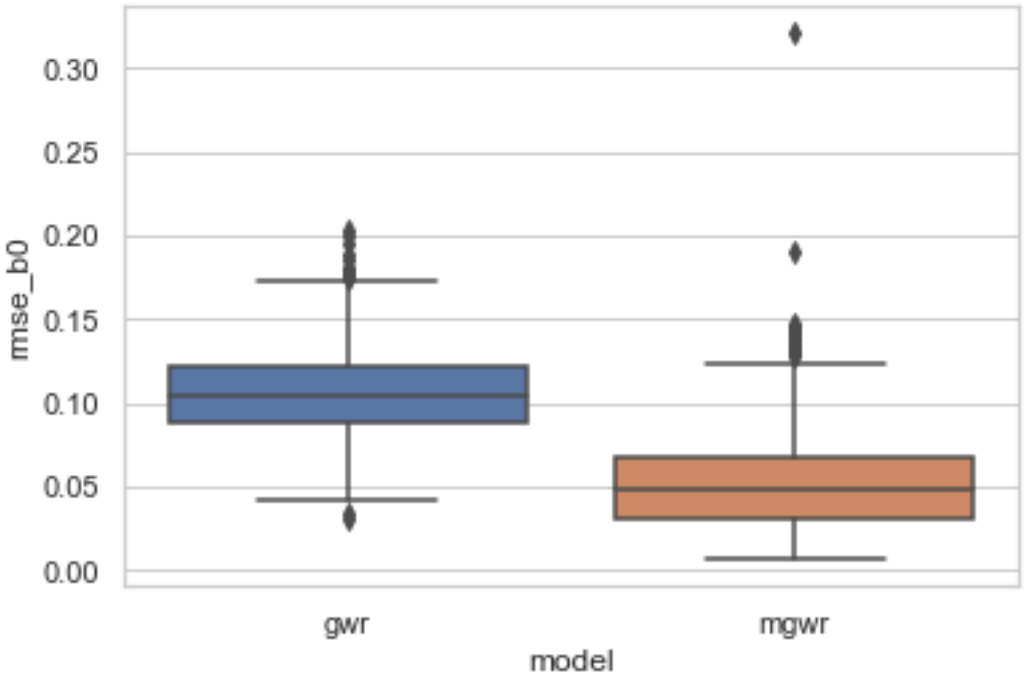
```
In [44]: B0_g=np.hsplit(gwr_params_mean,3)[0]
         B1_g=np.hsplit(gwr_params_mean,3)[1]
         B2_g=np.hsplit(gwr_params_mean,3)[2]

In [45]: B0_m=np.hsplit(mgwr_params_mean,3)[0]
         B1_m=np.hsplit(mgwr_params_mean,3)[1]
         B2_m=np.hsplit(mgwr_params_mean,3)[2]

In [46]: b0 = B0.reshape(-1,1)
         b1 = B1.reshape(-1,1)
         b2 = B2.reshape(-1,1)
```

0.0.9 B_0

```
In [156]: rmse_b0_m=[]
          for i in range(1000):
              rmse_b0_m.append(np.sqrt((np.sum((b0 - (np.hsplit(mgwr_params[i],3)[0]))**2))/625

          rmse_b0_g=[]
          for i in range(1000):
              rmse_b0_g.append(np.sqrt((np.sum((b0 - (np.hsplit(gwr_params[i],3)[0]))**2))/625)
```

```
In [157]: model=[]
          model = ['gwr']*1000
          model2 = ['mgwr']*1000
          model=model+model2

          rmse_b0 = rmse_b0_g+rmse_b0_m
          d = {"model":model,"rmse_b0":rmse_b0}
          df = pd.DataFrame(data=d)
```

```
In [158]: sns.set(style="whitegrid")
          ax = sns.boxplot(y=df['rmse_b0'],x=df['model'])
```



0.0.10 B_1

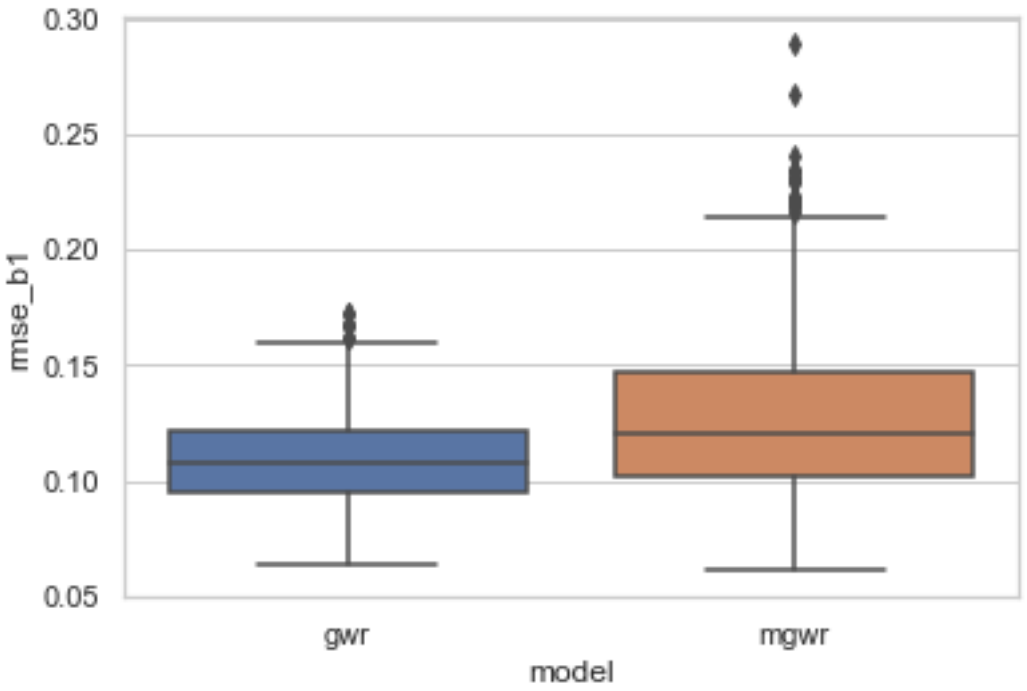
```
In [160]: rmse_b1_m=[]
          for i in range(1000):
```

```
        rmse_b1_m.append(np.sqrt((np.sum((b1 - (np.hsplit(mgwr_params[i],3)[1]))**2))/625
```

In [161]:
```
rmse_b1_g=[]
for i in range(1000):
    rmse_b1_g.append(np.sqrt((np.sum((b1 - (np.hsplit(gwr_params[i],3)[1]))**2))/625)
```

In [164]:
```
model=[]
model = ['gwr']*1000
model2 = ['mgwr']*1000
model=model+model2
rmse_b1=[]
rmse_b1 = rmse_b1_g+rmse_b1_m
d = {"model":model,"rmse_b1":rmse_b1}
df = pd.DataFrame(data=d)
```

In [165]:
```
sns.set(style="whitegrid")
ax = sns.boxplot(x=df['model'],y=df['rmse_b1'])
```



In [166]:
```
np.sqrt((np.sum((b1 - (np.hsplit(mgwr_params_mean,3)[1]))**2))/625)
```

Out[166]: 0.026800799078238548

In [167]:
```
np.sqrt((np.sum((b1 - (np.hsplit(gwr_params_mean,3)[1]))**2))/625)
```

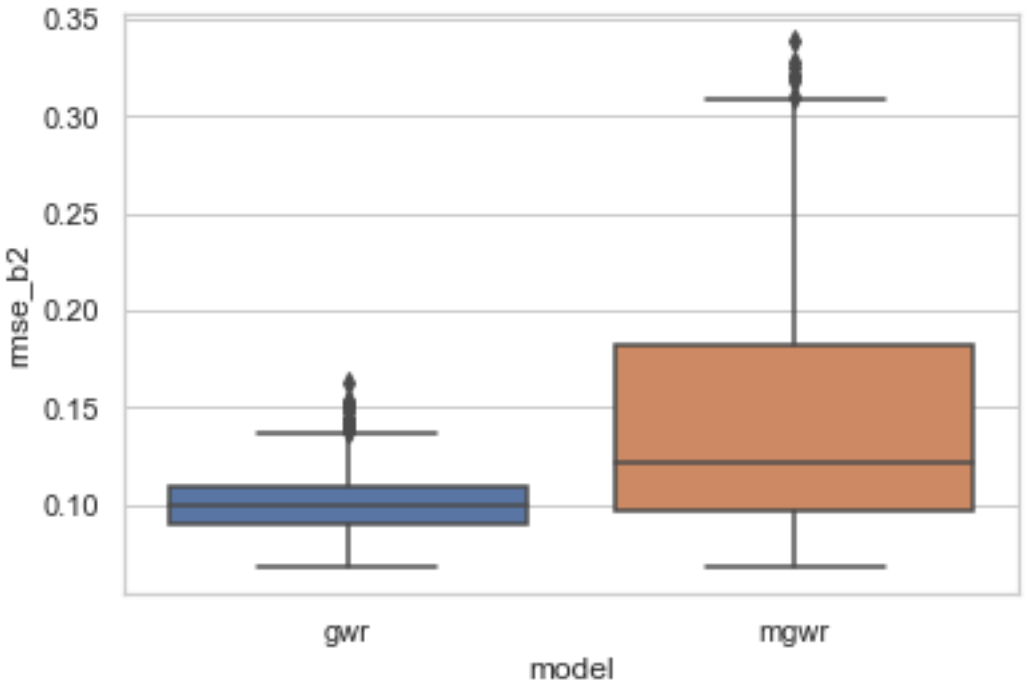
Out[167]: 0.06548680846620793

0.0.11 B_2

```
In [168]: rmse_b2_m=[]
          for i in range(1000):
              rmse_b2_m.append(np.sqrt((np.sum((b2 - np.hsplit(mgwr_params[i],3)[2])**2))/625))

          rmse_b2_g=[]
          for i in range(1000):
              rmse_b2_g.append(np.sqrt((np.sum((b2 - np.hsplit(gwr_params[i],3)[2])**2))/625))
```

```
In [170]: model=[]
          model = ['gwr']*1000
          model2 = ['mgwr']*1000
          model=model+model2
          rmse_b2=[]
          rmse_b2 = rmse_b2_g+rmse_b2_m
          d = {"model":model,"rmse_b2":rmse_b2}
          df = pd.DataFrame(data=d)
```

```
In [171]: sns.set(style="whitegrid")
          ax = sns.boxplot(y=df['rmse_b2'],x=df['model'])
```




```
In [172]: np.sqrt((np.sum((b2 - (np.hsplit(mgwr_params_mean,3)[2]))**2))/625)
```

```
Out[172]: 0.0524967462543234
```

```
In [173]: np.sqrt((np.sum((b2 - (np.hsplit(gwr_params_mean,3)[2]))**2))/625)

Out[173]: 0.06392889088066671
```