Binomial_MGWR_MonteCarlo_Results-we

May 14, 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/Binom
        import model_mc
        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 [5]: for i in range(0,500,10):
            p1 = pickle.load( open( "C:/Users/msachde1/Downloads/Research/Development/mgwr/note
            for j in range(10):
                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 [71]: 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 [72]: 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)
```

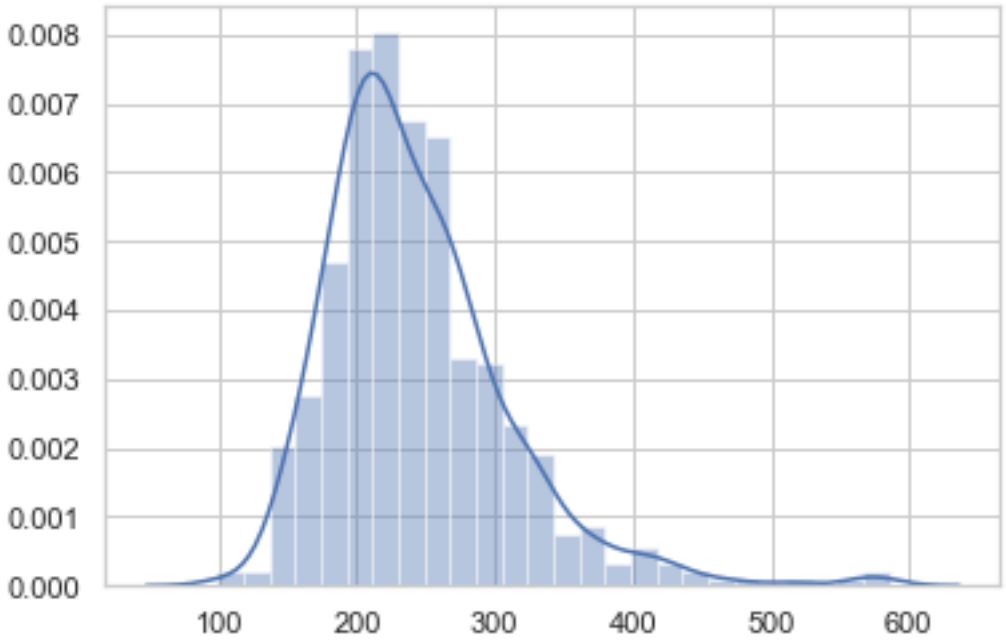
```
In [73]: x = np.linspace(0, 25, 25)
         y = np.linspace(25, 0, 25)
```

```
In [74]: x = np.linspace(0, 25, 25)
         y = np.linspace(25, 0, 25)
```

0.0.4 GWR bandwidth

```
In [75]: sns.distplot(gwr_bw)
```

Out[75]: <matplotlib.axes._subplots.AxesSubplot at 0x24e830ad0b8>



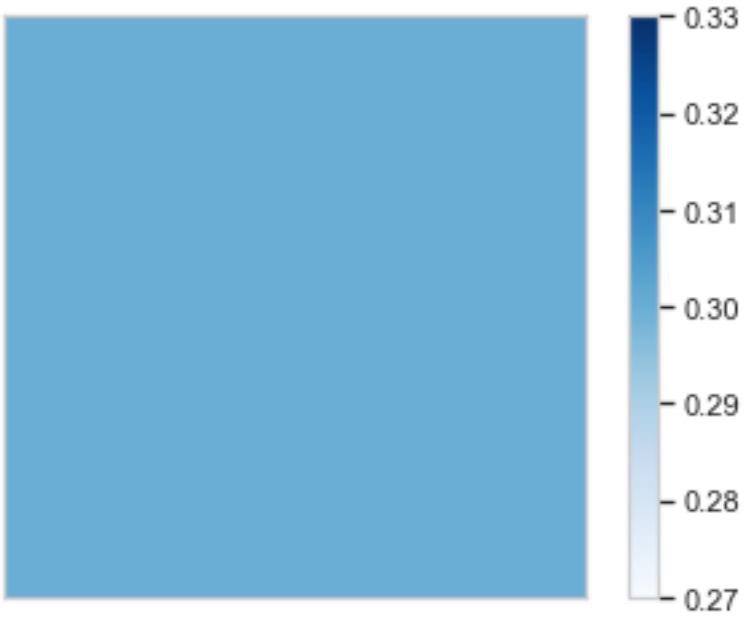
In [76]: np.mean(gwr_bw)

Out[76]: 243.354

0.0.5 MGWR bandwidths

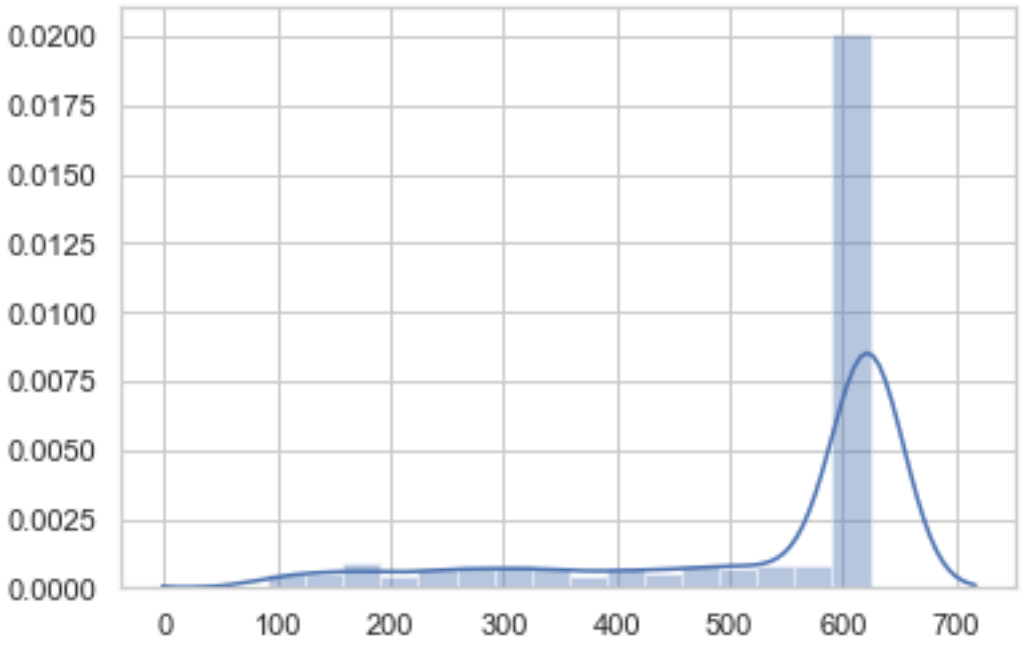
```
In [77]: plt.imshow(B0, extent=[0,10, 0, 10], origin='lower',cmap='Blues')
         plt.colorbar()
         plt.axis(aspect='image')
         plt.xticks([])
         plt.yticks([])
```

Out[77]: ([], <a list of 0 Text yticklabel objects>)

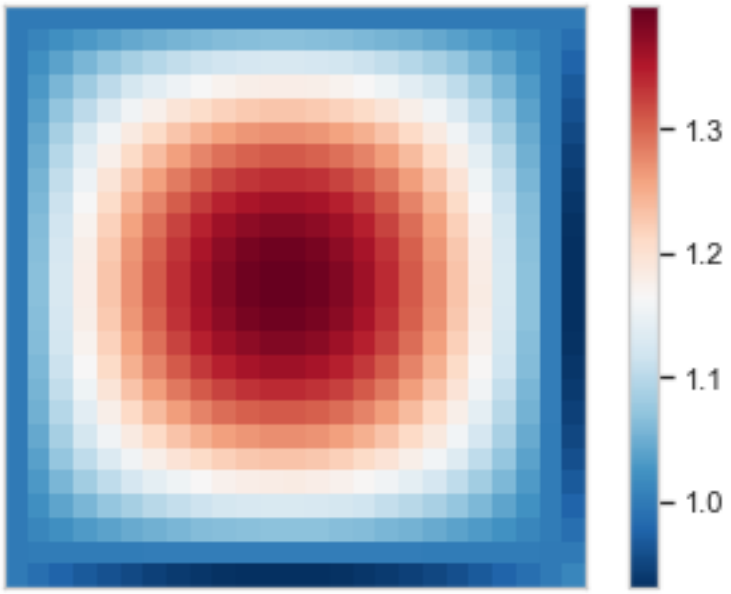


In [78]: sns.distplot(mgwr_bw0)

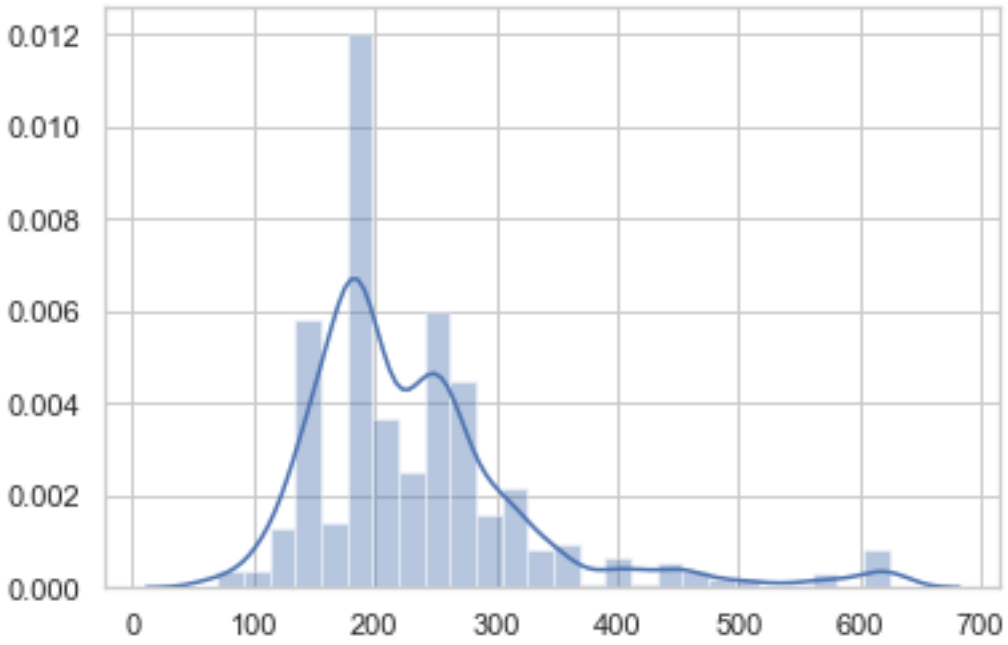
Out[78]: <matplotlib.axes._subplots.AxesSubplot at 0x24e837a1d30>



```
In [79]: np.mean(mgwr_bw0)

Out[79]: 532.386

In [80]: plt.imshow(B1, extent=[0,25, 0, 25], origin='lower',cmap='RdBu_r')
         plt.colorbar()
         plt.axis(aspect='image')
         plt.xticks([])
         plt.yticks([])

Out[80]: ([], <a list of 0 Text yticklabel objects>)
```

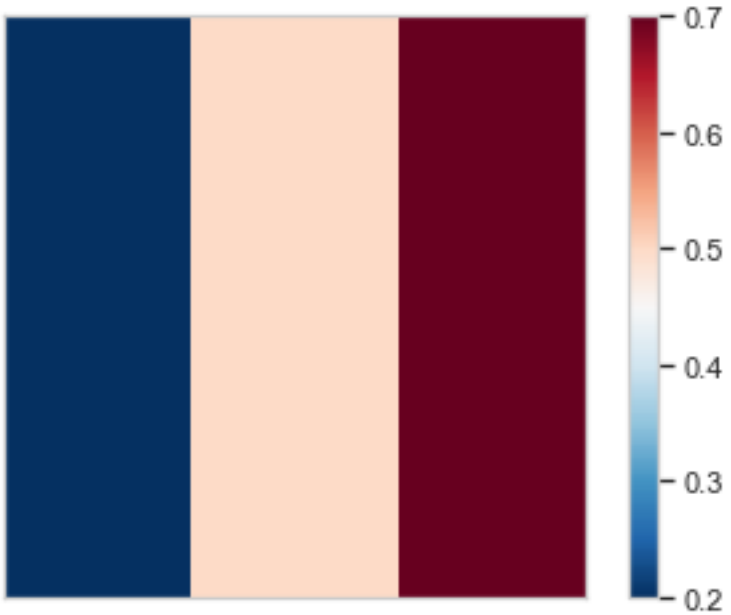



```
In [81]: np.mean(mgwr_bw1)

Out[81]: 234.168

In [82]: sns.distplot(mgwr_bw1)

Out[82]: <matplotlib.axes._subplots.AxesSubplot at 0x24e83b4fc50>
```




```
In [83]: plt.imshow(B2, extent=[0,25, 0, 25], origin='lower',cmap='RdBu_r')
         plt.colorbar()
         plt.axis(aspect='image')
         plt.xticks([])
         plt.yticks([])
```

```
Out[83]: ([], <a list of 0 Text yticklabel objects>)
```

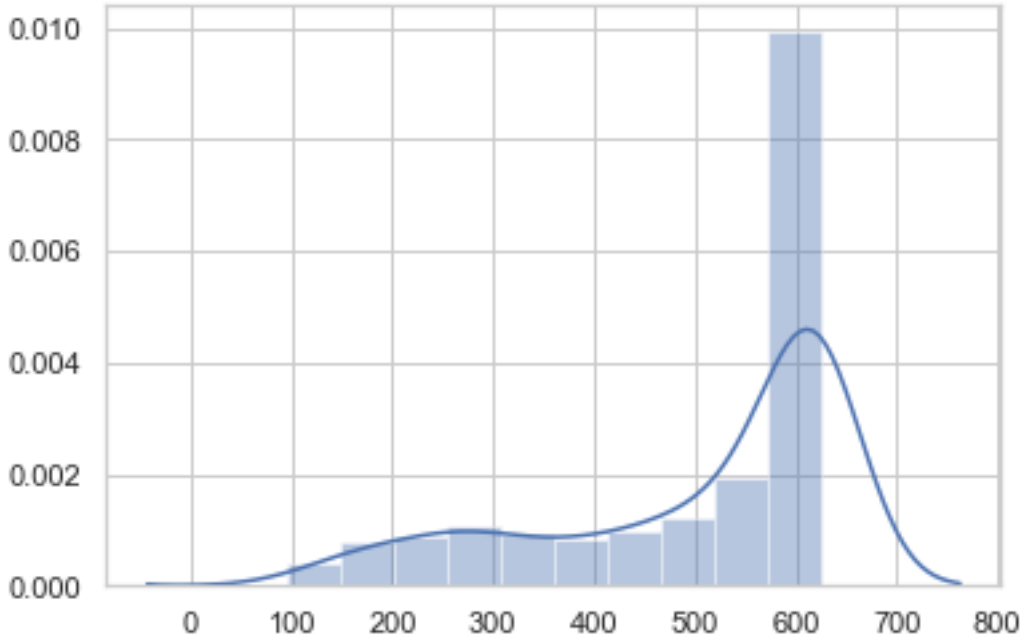


```
In [84]: np.mean(mgwr_bw2)
```

```
Out[84]: 502.406
```

```
In [85]: sns.distplot(mgwr_bw2)
```

```
Out[85]: <matplotlib.axes._subplots.AxesSubplot at 0x24e834a5f60>
```



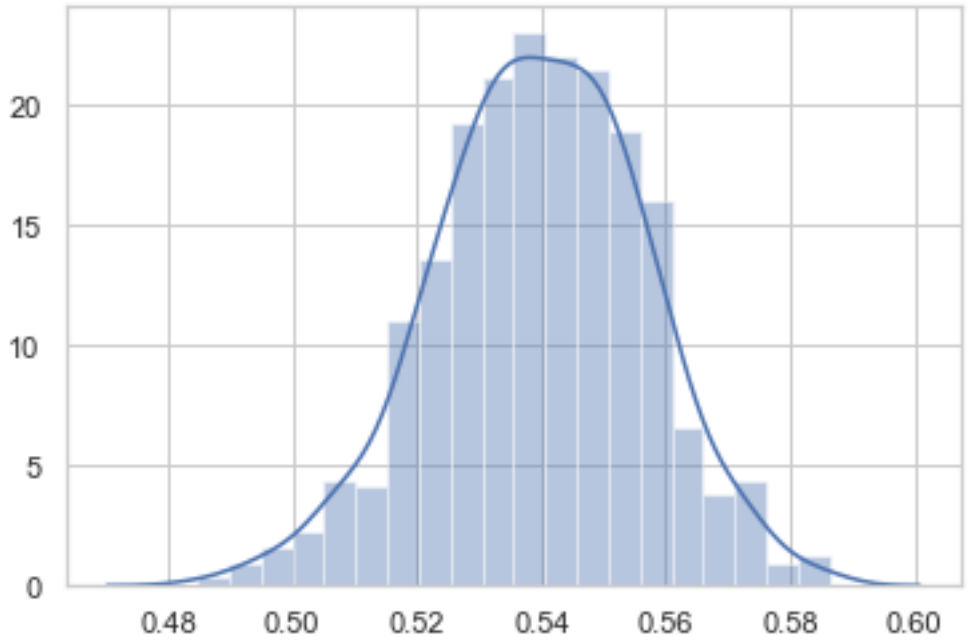

```
In [86]: np.mean(mgwr_bw0),np.mean(mgwr_bw1),np.mean(mgwr_bw2)
```

```
Out[86]: (532.386, 234.168, 502.406)
```

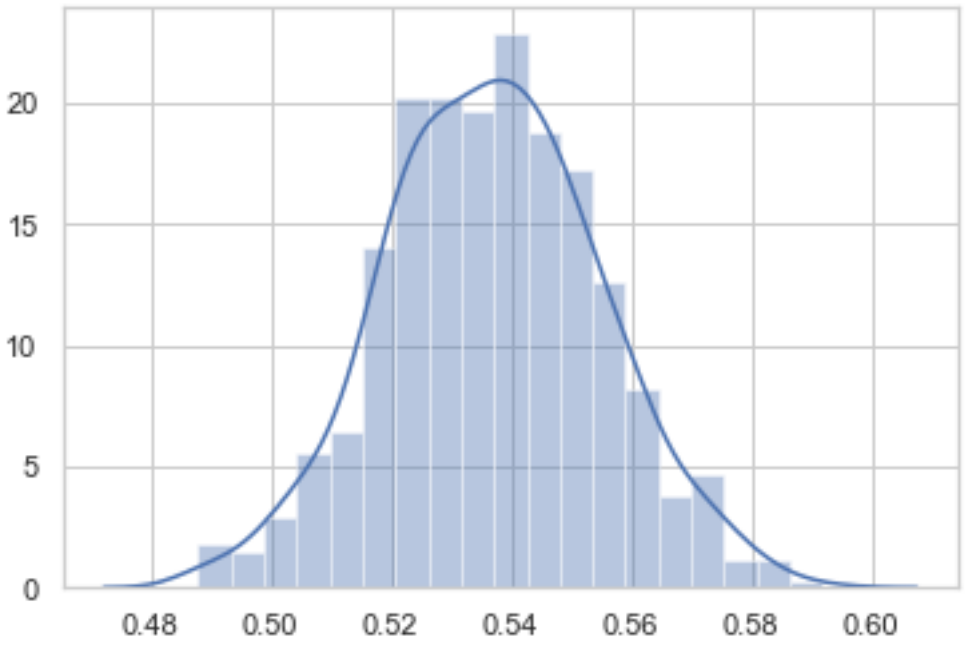
0.0.6 AIC, AICc, BIC check

```
In [87]: mgwr_aicc=[]
         gwr_aicc=[]
         mgwr_bic=[]
         gwr_bic=[]
         mgwr_aic=[]
         gwr_aic=[]
         mgwr_params=[]
         gwr_params=[]
         mgwr_predy=[]
         gwr_predy=[]
```

```
In [88]: for i in range(0,500,10):
             p1 = pickle.load( open( "C:/Users/msachde1/Downloads/Research/Development/mgwr/no
             for j in range(10):
                 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 [89]: sns.distplot(np.mean(gwr_predy,axis=0))

Out[89]: <matplotlib.axes._subplots.AxesSubplot at 0x24e83099898>
```

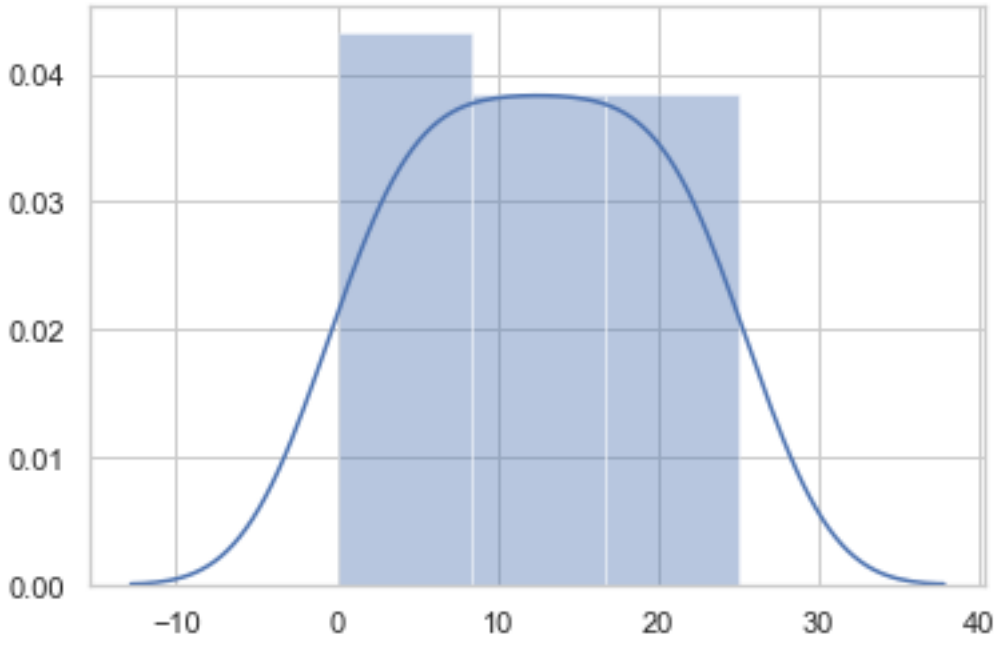



```
In [90]: sns.distplot(np.mean(mgwr_predy,axis=0))

Out[90]: <matplotlib.axes._subplots.AxesSubplot at 0x24e81caef28>
```

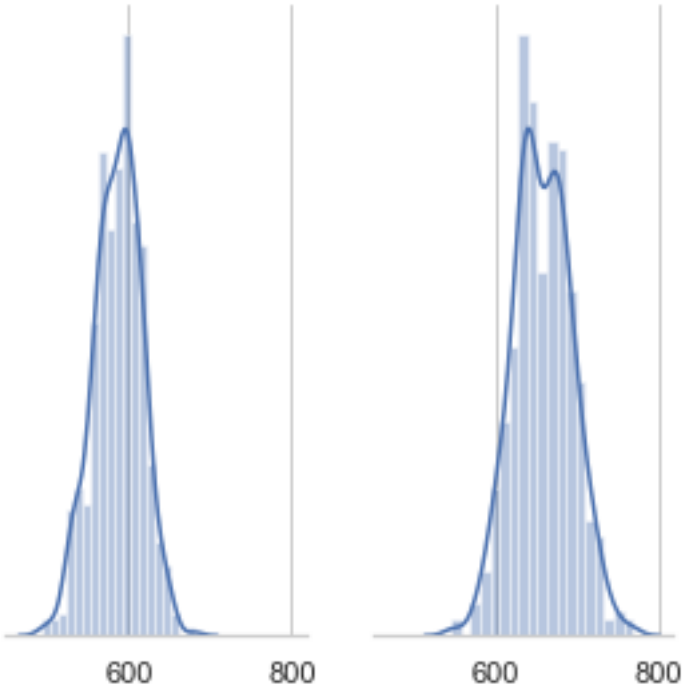



```
In [91]: sns.distplot(y)
```

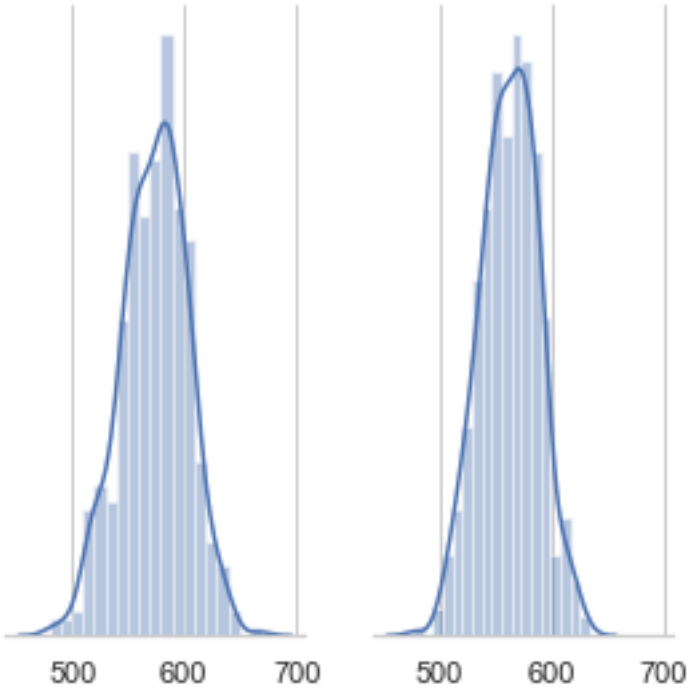
```
Out[91]: <matplotlib.axes._subplots.AxesSubplot at 0x24e81e7e4a8>
```



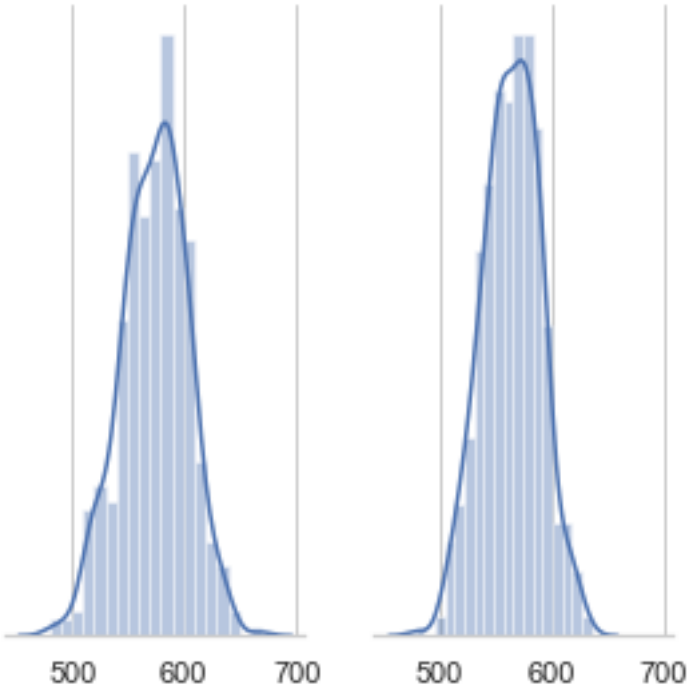
In [92]:
```
f, axes = plt.subplots(1, 2, figsize=(4, 4), sharex=True)
sns.despine(left=True)
sns.distplot(mgwr_bic,ax=axes[0])
sns.distplot(gwr_bic,ax=axes[1])

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



In [93]:
```
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 [94]: 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 [95]: np.mean(mgwr_aicc), np.mean(gwr_aicc)
```

```
Out[95]: (573.0542142681326, 564.1109593727111)
```

```
In [96]: np.mean(mgwr_aic), np.mean(gwr_aic)
```

```
Out[96]: (573.0155189048916, 562.4075174365614)
```

```
In [97]: np.mean(mgwr_bic), np.mean(gwr_bic)
```

```
Out[97]: (586.3381072542409, 657.4481261493207)
```

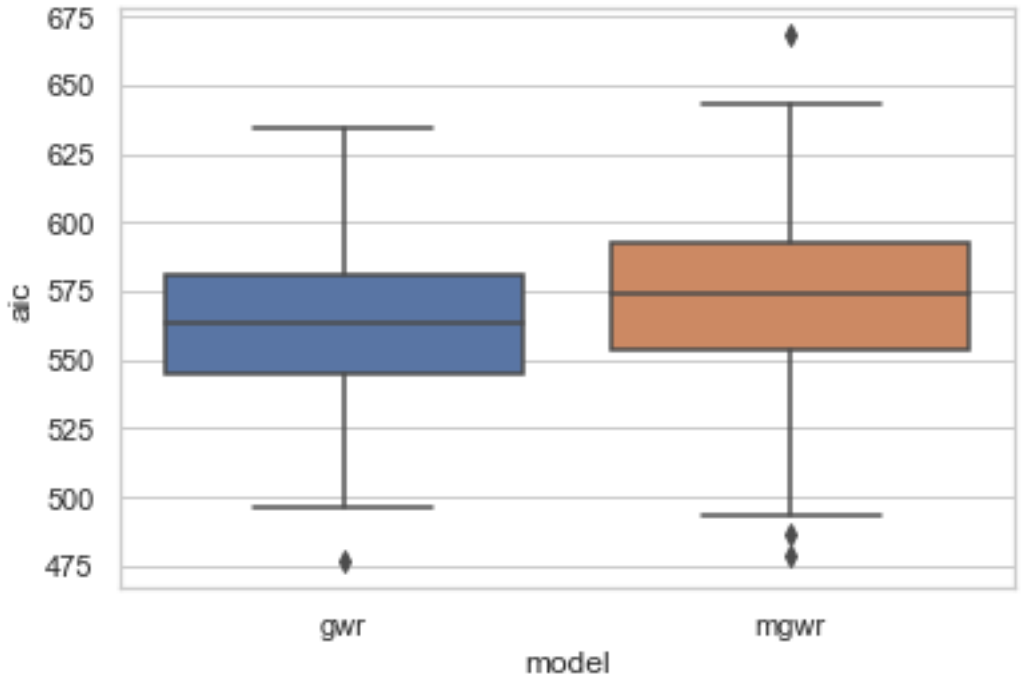
AIC, AICc, BIC Boxplots for comparison

```
In [98]: model=[]
         model = ['gwr']*500
         model2 = ['mgwr']*500
         model=model+model2
```

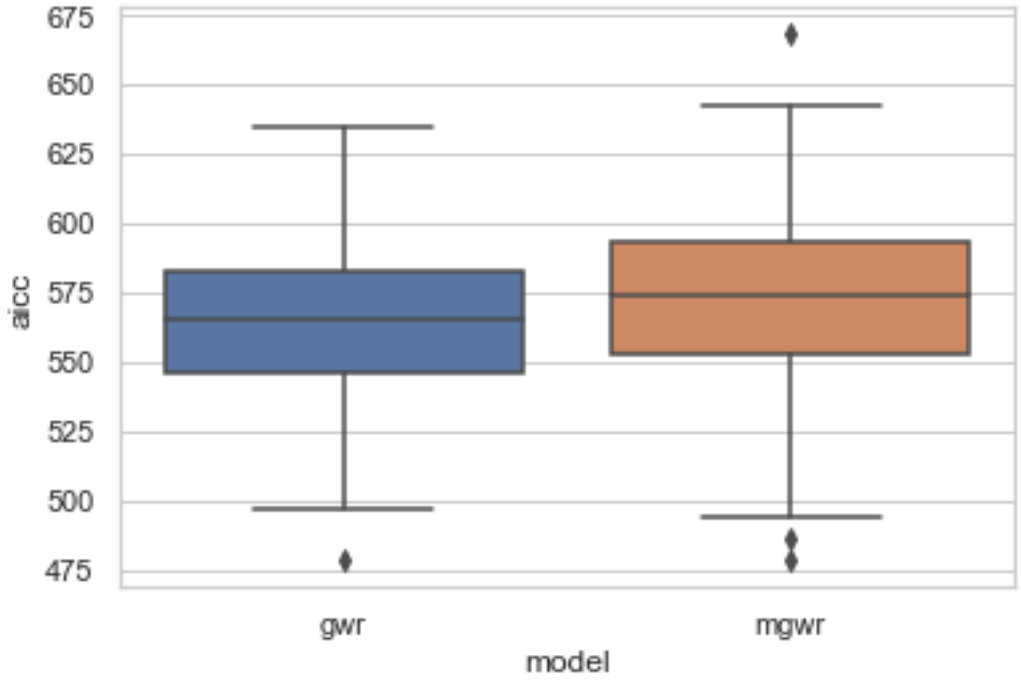
```
In [99]: aic=[]
         aic=gwr_aic
         aic=aic+mgwr_aic
```

```
In [100]: aicc=[]
          aicc=gwr_aicc
          aicc=aicc+mgwr_aicc
```

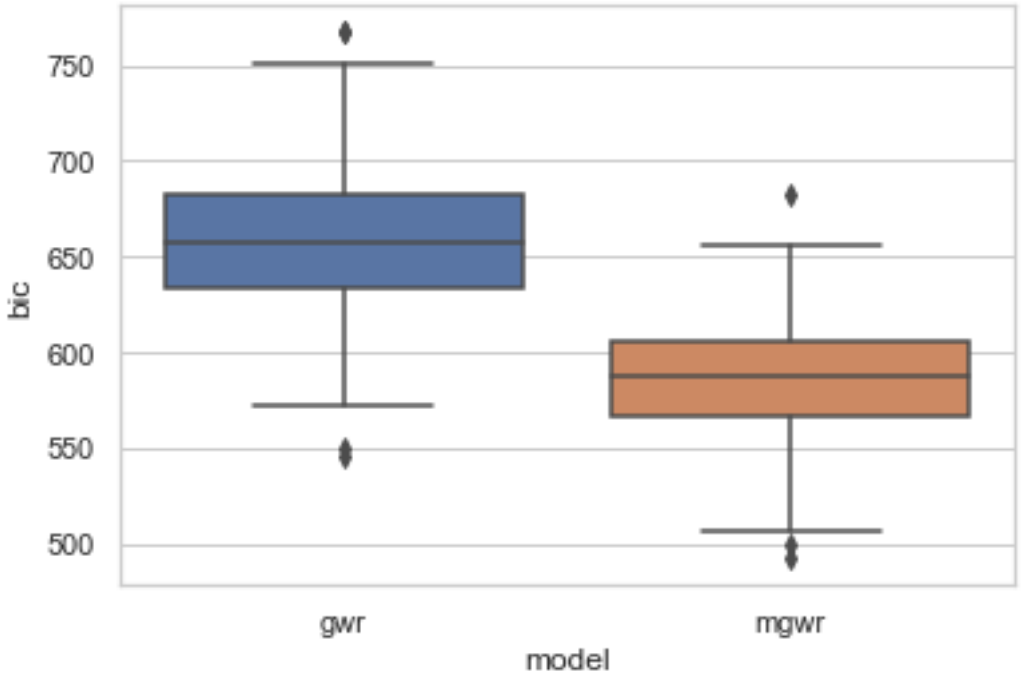
```
In [101]: bic=[]
          bic=gwr_bic
          bic=bic+mgwr_bic

In [102]: d = {'aic':aic,'bic':bic,'aicc':aicc,'model':model}

In [103]: df=pd.DataFrame(data=d)

In [104]: sns.set(style="whitegrid")
          ax = sns.boxplot(y=df['aic'],x=df['model'])
```




```
In [105]: sns.set(style="whitegrid")
          ax = sns.boxplot(y=df['aicc'],x=df['model'])
```



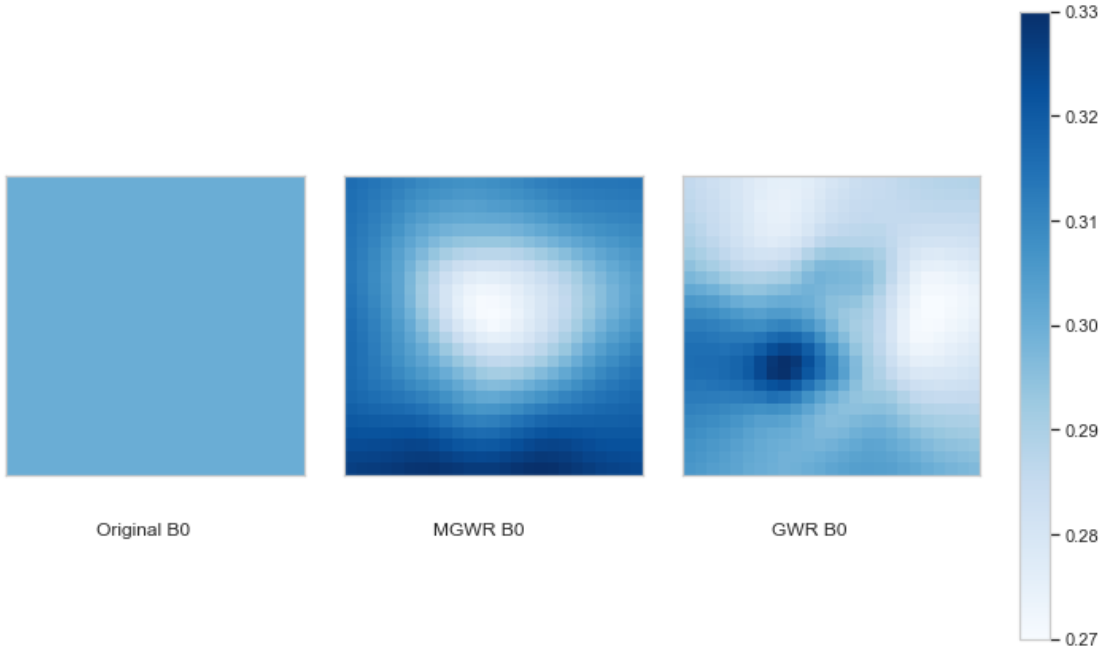
In [106]:
```
sns.set(style="whitegrid")
ax = sns.boxplot(y=df['bic'],x=df['model'])
```



0.0.7 Parameter comparison from MGWR and GWR

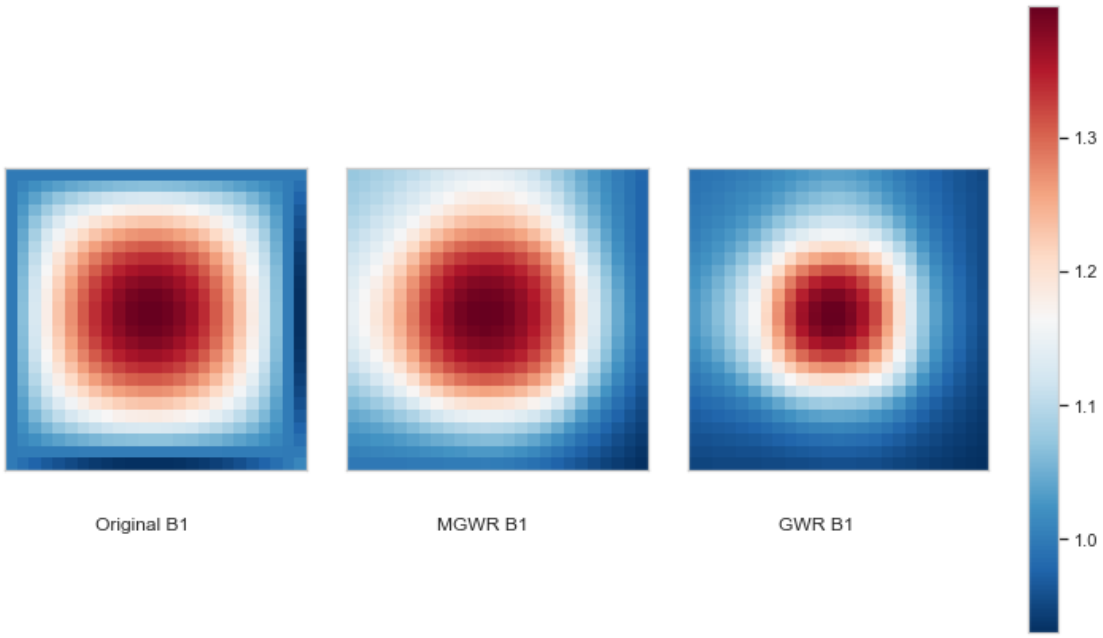
```
In [107]: mgwr_params_mean=np.mean(mgwr_params,axis=0)
          gwr_params_mean=np.mean(gwr_params,axis=0)

In [108]: gwr_params_mean

Out[108]: array([[0.31469142, 1.61932981, 0.24680036],
                 [0.31376851, 1.62072329, 0.25249583],
                 [0.31251579, 1.62234972, 0.25957388],
                 ...,
                 [0.29442015, 1.71479871, 0.64257558],
                 [0.2947018 , 1.67897208, 0.64781585],
                 [0.29479386, 1.64721497, 0.65266   ]])

In [109]: 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 [110]: 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 [111]: B0_mgwr=B0_mgwr.reshape(25,25)
          B1_mgwr=B1_mgwr.reshape(25,25)
          B2_mgwr=B2_mgwr.reshape(25,25)

In [112]: np.mean(B0_mgwr),np.mean(B0_gwr),np.mean(B0)

Out[112]: (0.3162876373767867, 0.3000642017402459, 0.3)

In [113]: B0_gwr=B0_gwr.reshape(25,25)
          B1_gwr=B1_gwr.reshape(25,25)
          B2_gwr=B2_gwr.reshape(25,25)

In [114]: 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()
```



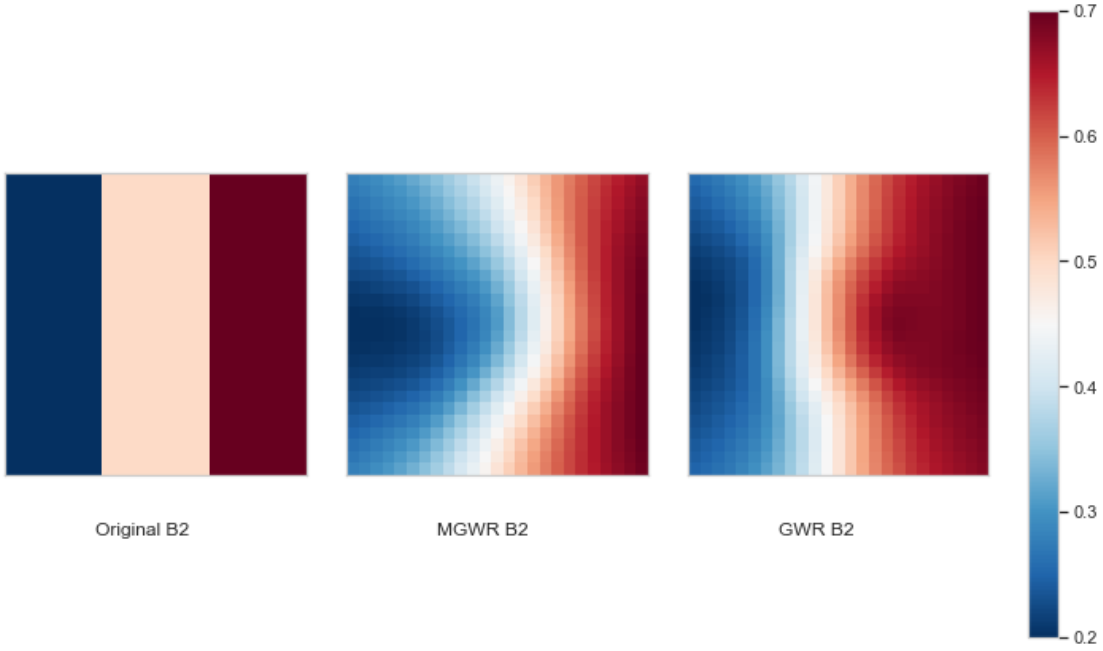
In [115]:
```
fig, (ax, ax2,ax3, cax) = plt.subplots(ncols=4,figsize=(10,6),
                         gridspec_kw={"width_ratios":[1,1,1, 0.1],"height_ratios":[1]})
fig.subplots_adjust(wspace=0.3)
im = ax.imshow(B1, extent=[0,10, 0, 10], origin='lower',cmap='RdBu_r')
ax.text(3, -2, 'Original B1')
im2  = 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 [116]:
```
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 [52]: 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 [53]: 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]
```

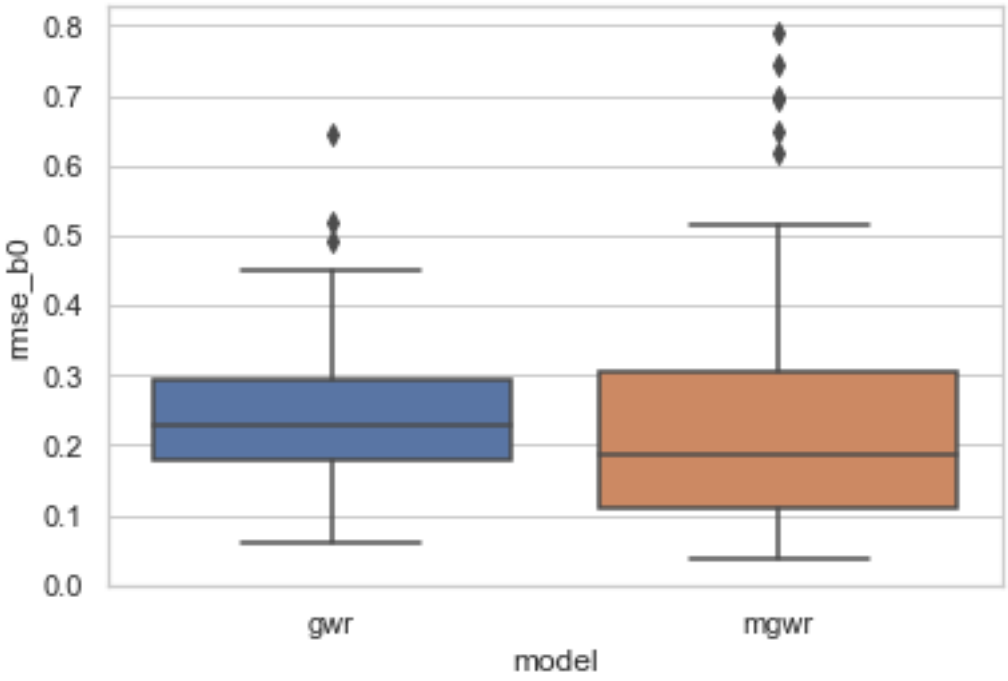
```
In [54]: b0 = B0.reshape(-1,1)
         b1 = B1.reshape(-1,1)
         b2 = B2.reshape(-1,1)
```

0.0.9 B_0

```
In [55]: rmse_b0_m=[]
         for i in range(100):
             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(100):
             rmse_b0_g.append(np.sqrt((np.sum((b0 - (np.hsplit(gwr_params[i],3)[0]))**2))/625))
```

```
In [56]: model=[]
         model = ['gwr']*100
         model2 = ['mgwr']*100
         model=model+model2

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

```
In [57]: sns.set(style="whitegrid")
         ax = sns.boxplot(y=df['rmse_b0'],x=df['model'])
```



0.0.10 B_1

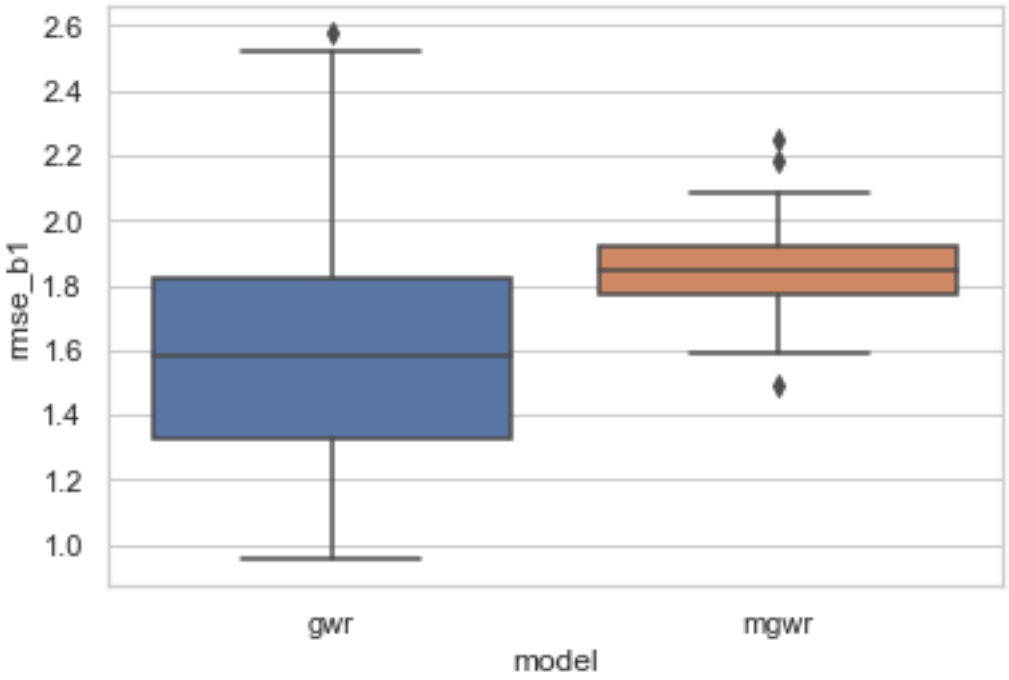
```
In [58]: rmse_b1_m=[]
         for i in range(100):
```

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

In [59]:
```
rmse_b1_g=[]
for i in range(100):
    rmse_b1_g.append(np.sqrt((np.sum((b1 - (np.hsplit(gwr_params[i],3)[1]))**2))/625))
```

In [60]:
```
model=[]
model = ['gwr']*100
model2 = ['mgwr']*100
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 [61]:
```
sns.set(style="whitegrid")
ax = sns.boxplot(x=df['model'],y=df['rmse_b1'])
```



In [62]:
```
np.sqrt((np.sum((b1 - (np.hsplit(mgwr_params_mean,3)[1]))**2))/625)
```

Out[62]:
```
1.8237125254924693
```

In [63]:
```
np.sqrt((np.sum((b1 - (np.hsplit(gwr_params_mean,3)[1]))**2))/625)
```

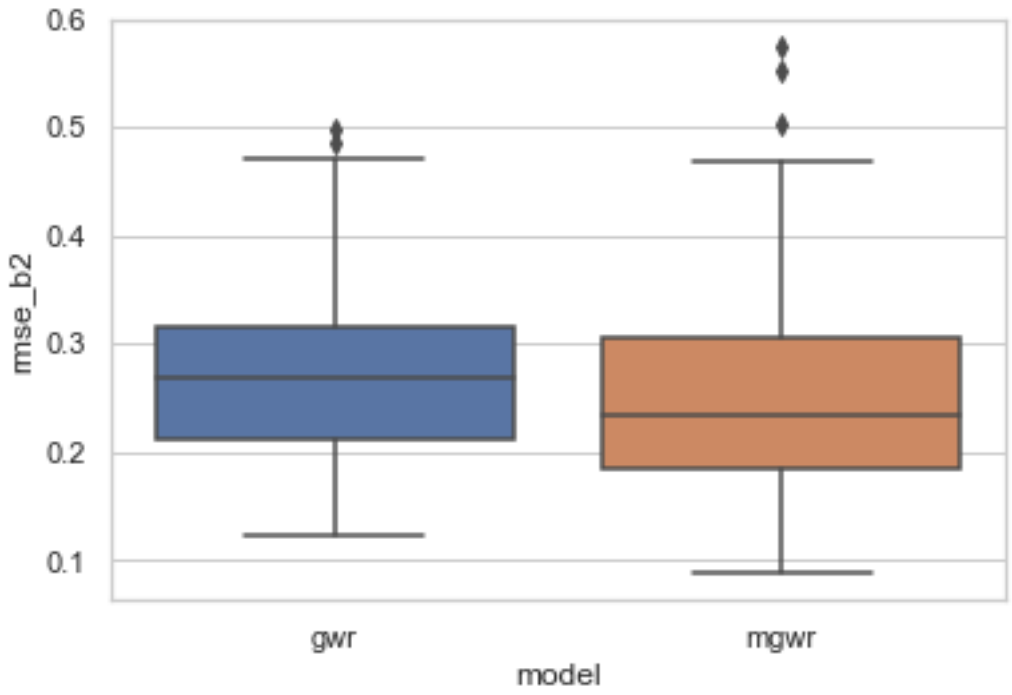
Out[63]:
```
1.5244657525329972
```

0.0.11 B_2

```
In [64]: rmse_b2_m=[]
         for i in range(100):
             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(100):
             rmse_b2_g.append(np.sqrt((np.sum((b2 - np.hsplit(gwr_params[i],3)[2])**2))/625))
```

```
In [65]: model=[]
         model = ['gwr']*100
         model2 = ['mgwr']*100
         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 [66]: sns.set(style="whitegrid")
         ax = sns.boxplot(y=df['rmse_b2'],x=df['model'])
```




```
In [67]: np.sqrt((np.sum((b2 - (np.hsplit(mgwr_params_mean,3)[2]))**2))/625)
```

```
Out[67]: 0.10002993807213198
```

```
In [68]: np.sqrt((np.sum((b2 - (np.hsplit(gwr_params_mean,3)[2]))**2))/625)

Out[68]: 0.07791108751758576
```