Binomial_MGWR_MonteCarlo_Results

May 14, 2020

Notebook Outline:

- Section 0.0.1
- Section 0.0.2
- Section 0.0.3
- Section 0.0.4
- Section 0.0.5
- Section 0.0.6
    - Section 0.0.6
- Section 0.0.7

Monte Carlo experiment code can be found in path mgwr/notebooks/Binomial_MC_script/ (branch - gsoc19)

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/Binomi
        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\i
  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)
```

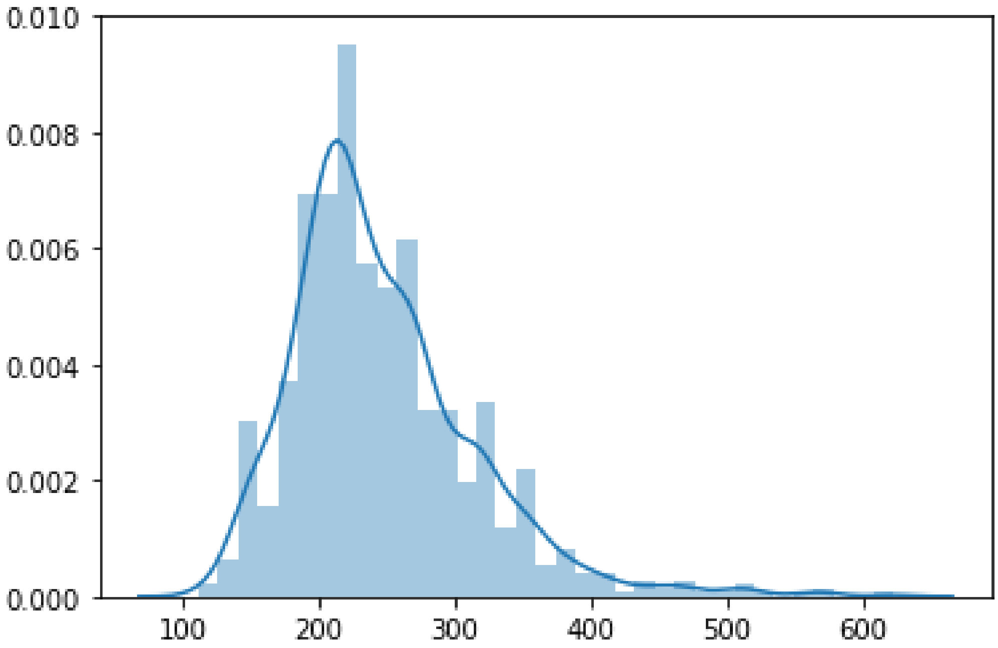
```
In [6]: x = np.linspace(0, 25, 25)
        y = np.linspace(25, 0, 25)
```

```
In [7]: x = np.linspace(0, 25, 25)
        y = np.linspace(25, 0, 25)
```

0.0.4 GWR bandwidth

```
In [8]: sns.distplot(gwr_bw)
```

```
Out[8]: <matplotlib.axes._subplots.AxesSubplot at 0x1460f35f470>
```



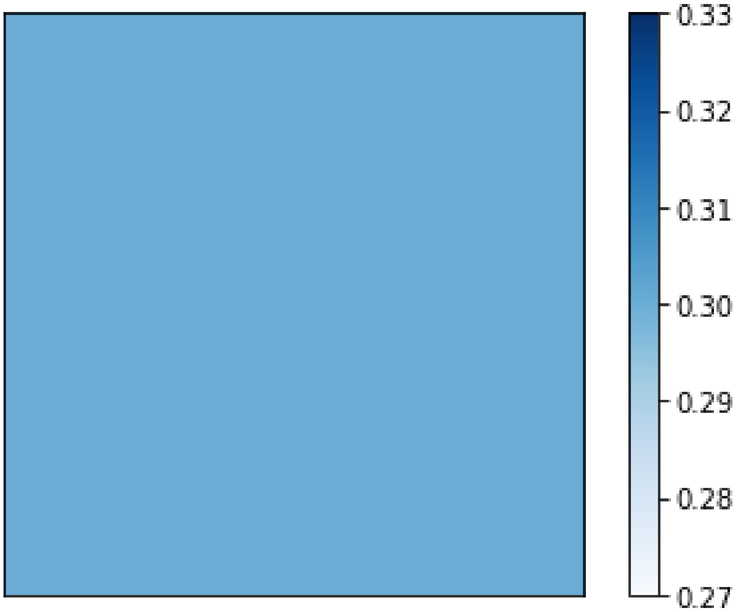
```
In [9]: np.mean(gwr_bw)
```

```
Out[9]: 246.295
```

0.0.5 MGWR bandwidths

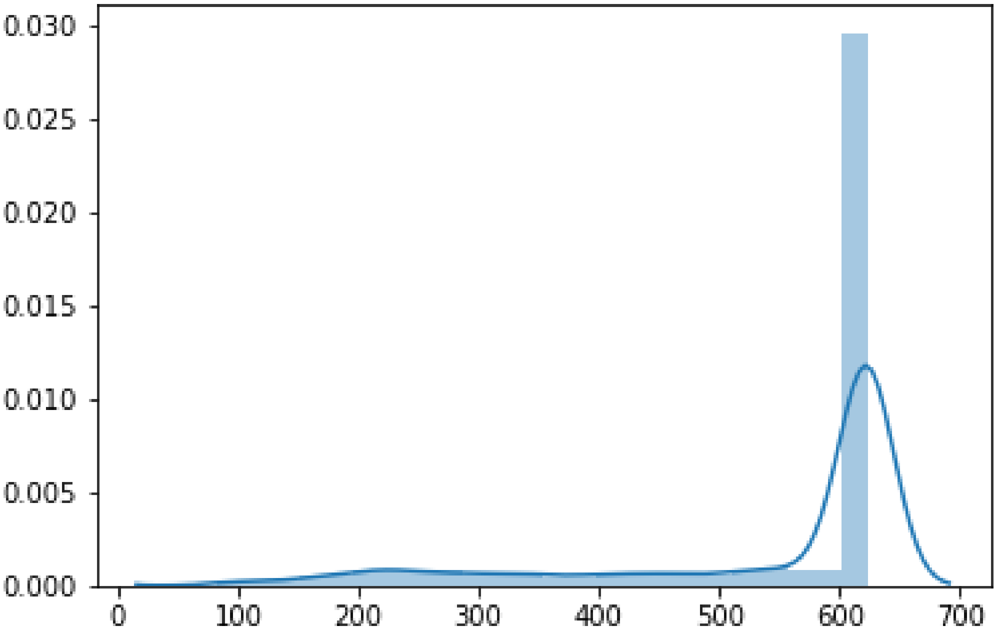
```
In [10]: plt.imshow(B0, extent=[0,10, 0, 10], origin='lower',cmap='Blues')
         plt.colorbar()
         plt.axis(aspect='image')
         plt.xticks([])
         plt.yticks([])
```

```
Out[10]: ([], <a list of 0 Text yticklabel objects>)
```



```
In [11]: sns.distplot(mgwr_bw0)
```

```
Out[11]: <matplotlib.axes._subplots.AxesSubplot at 0x1460f778438>
```

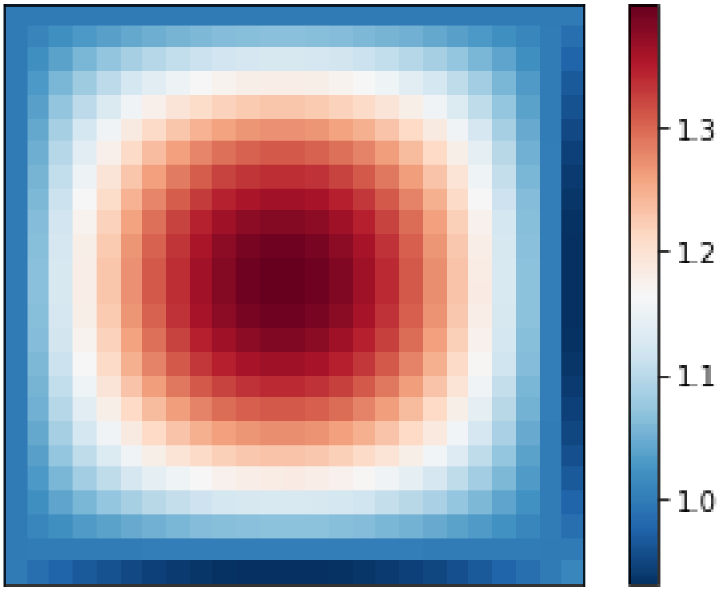


```
In [12]: np.mean(mgwr_bw0)
```

```
Out[12]: 538.887
```

```
In [13]: plt.imshow(B1, extent=[0,25, 0, 25], origin='lower',cmap='RdBu_r')
         plt.colorbar()
         plt.axis(aspect='image')
         plt.xticks([])
         plt.yticks([])
```

```
Out[13]: ([], <a list of 0 Text yticklabel objects>)
```

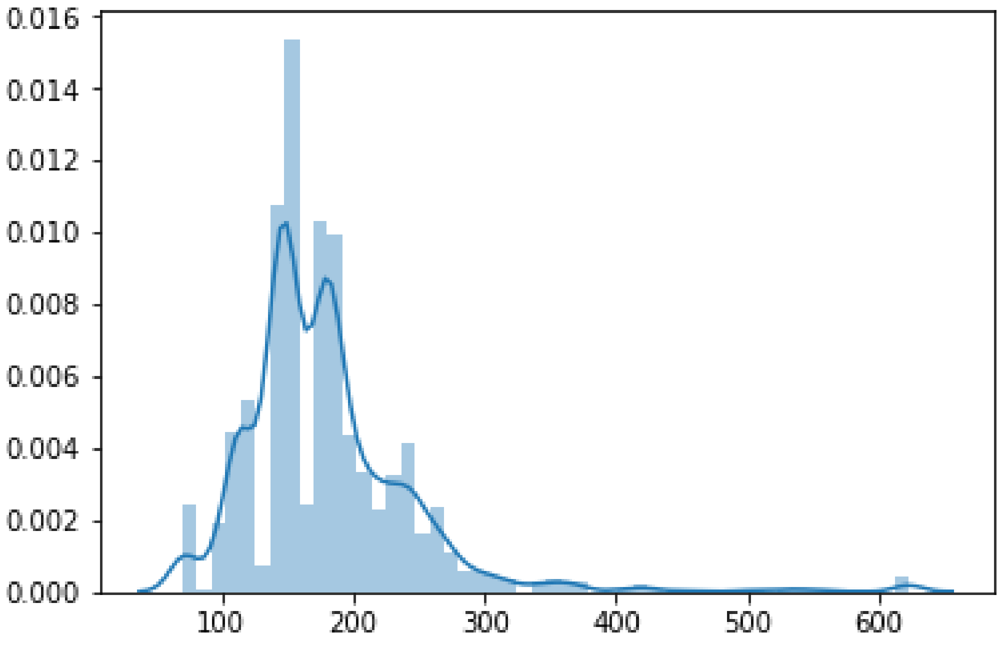


```
In [14]: np.mean(mgwr_bw1)
```

```
Out[14]: 179.233
```

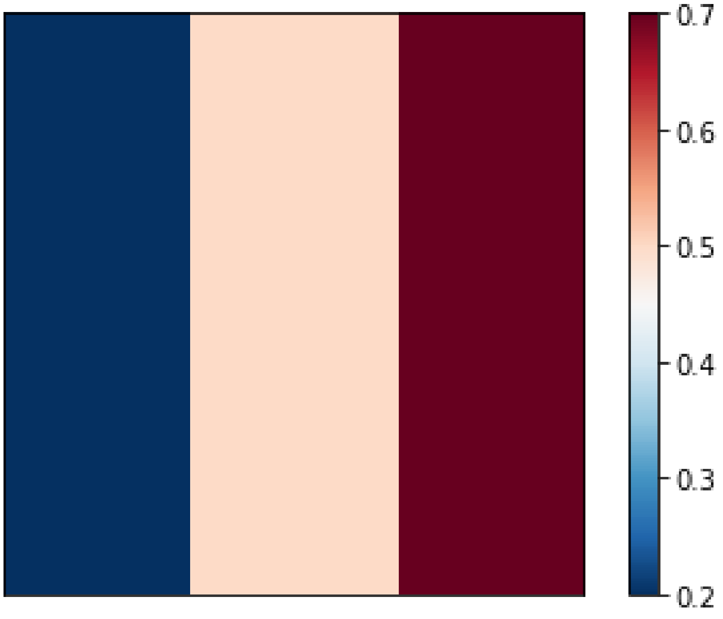
```
In [15]: sns.distplot(mgwr_bw1)
```

```
Out[15]: <matplotlib.axes._subplots.AxesSubplot at 0x1460f81a2e8>
```

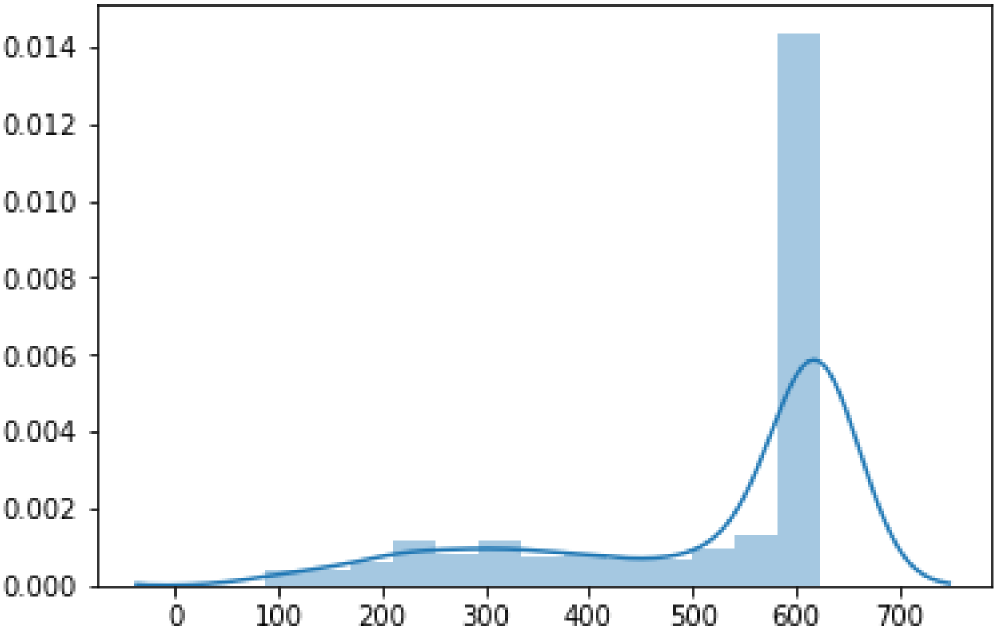


```
In [16]: plt.imshow(B2, extent=[0,25, 0, 25], origin='lower',cmap='RdBu_r')
         plt.colorbar()
         plt.axis(aspect='image')
         plt.xticks([])
         plt.yticks([])
```

```
Out[16]: ([], <a list of 0 Text yticklabel objects>)
```



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

Out[17]: 514.968

In [18]: sns.distplot(mgwr_bw2)

Out[18]: <matplotlib.axes._subplots.AxesSubplot at 0x1460f963128>
```



```
In [19]: np.mean(mgwr_bw0),np.mean(mgwr_bw1),np.mean(mgwr_bw2)

Out[19]: (538.887, 179.233, 514.968)
```

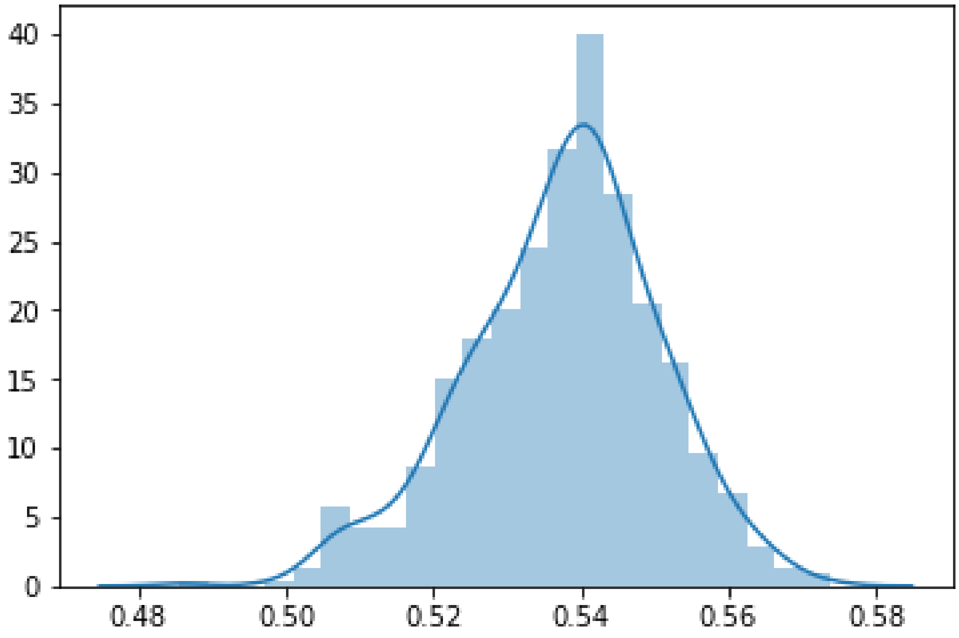
0.0.6 AIC, AICc, BIC check

```
In [20]: mgwr_aicc=[]
         gwr_aicc=[]
         mgwr_bic=[]
         gwr_bic=[]
         mgwr_aic=[]
         gwr_aic=[]
         mgwr_params=[]
         gwr_params=[]
         mgwr_predy=[]
         gwr_predy=[]
```

```
In [21]: for i in range(0,1000,50):
             p1 = pickle.load( open( "C:/Users/msachde1/Downloads/Research/Development/mgwr/no
             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])
```

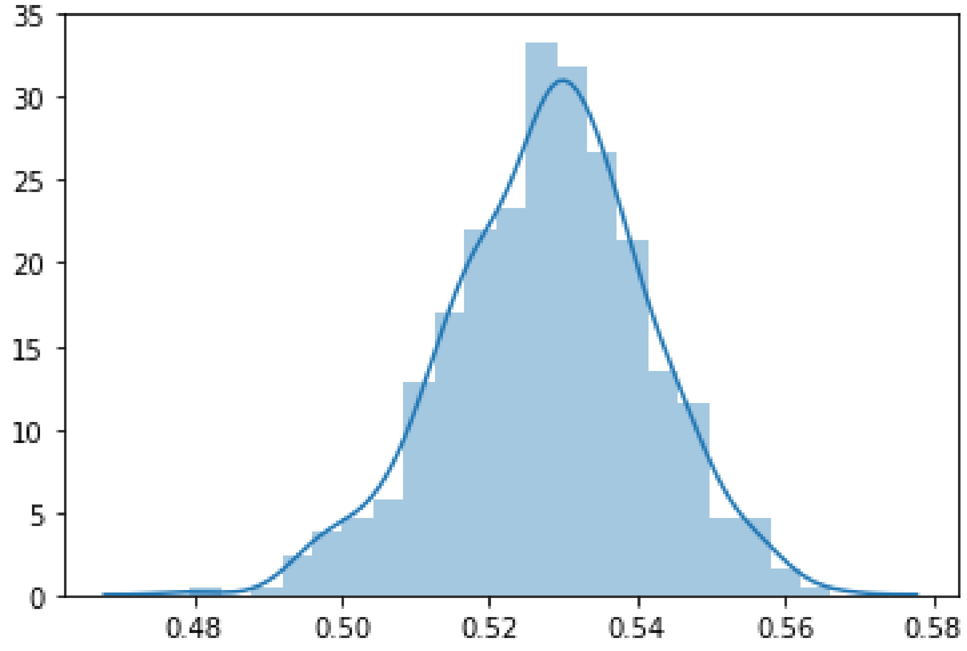
```
In [22]: sns.distplot(np.mean(gwr_predy,axis=0))
```

```
Out[22]: <matplotlib.axes._subplots.AxesSubplot at 0x146109b99b0>
```

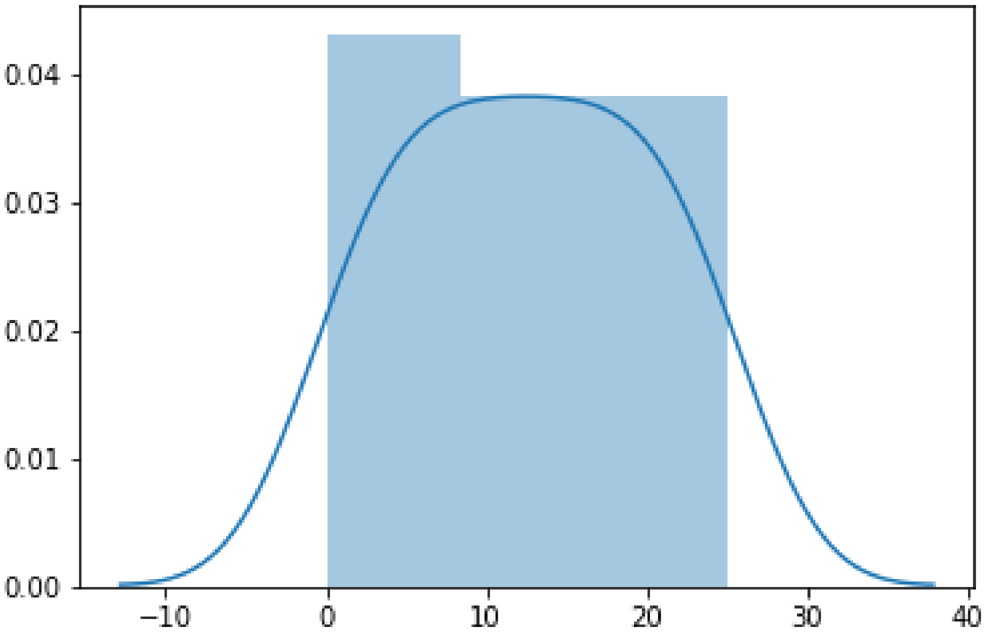


```
In [23]: sns.distplot(np.mean(mgwr_predy,axis=0))
```

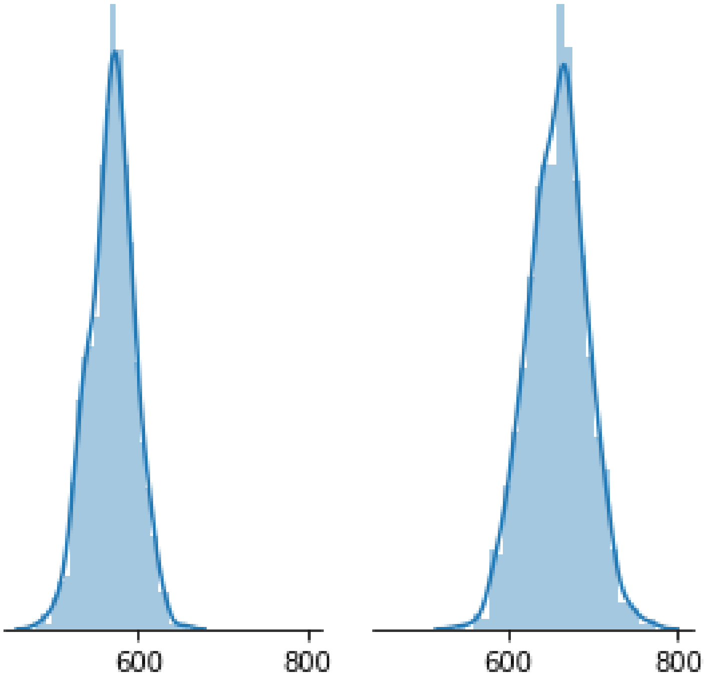
```
Out[23]: <matplotlib.axes._subplots.AxesSubplot at 0x14612a449b0>
```



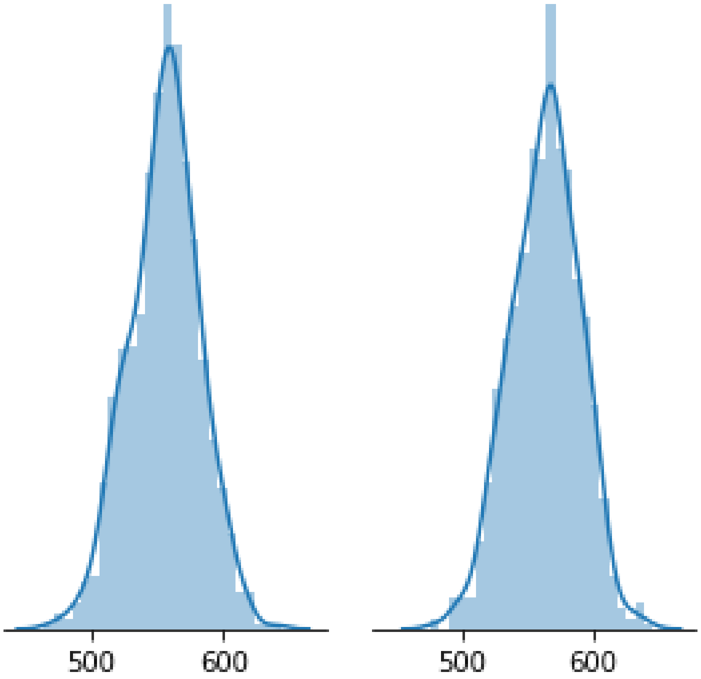
```
In [24]: sns.distplot(y)

Out[24]: <matplotlib.axes._subplots.AxesSubplot at 0x1460f8deb70>
```



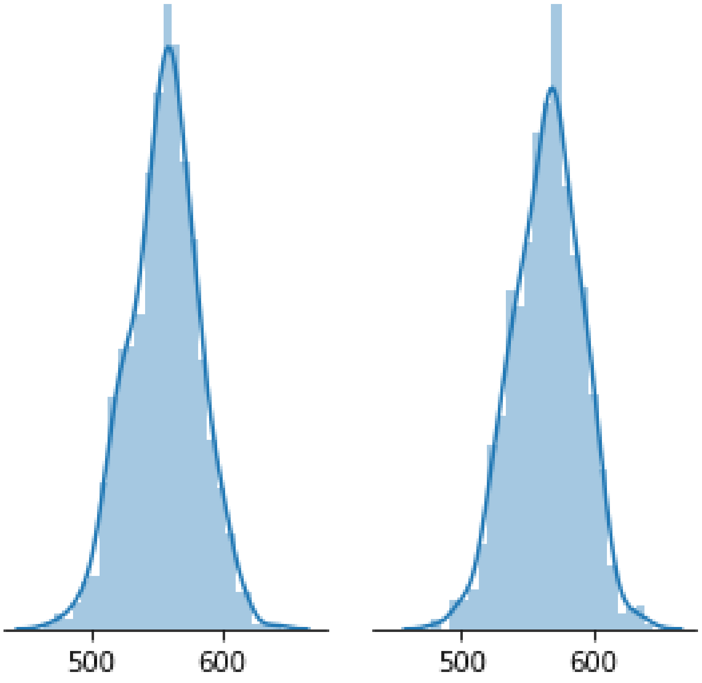
```
In [25]: f, axes = plt.subplots(1, 2, figsize=(4, 4), sharex=True)
         sns.despine(left=True)
         sns.distplot(mgwr_bic,ax=axes[0])
         sns.distplot(gwr_bic,ax=axes[1])

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



```
In [26]: 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 [27]: 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()
```

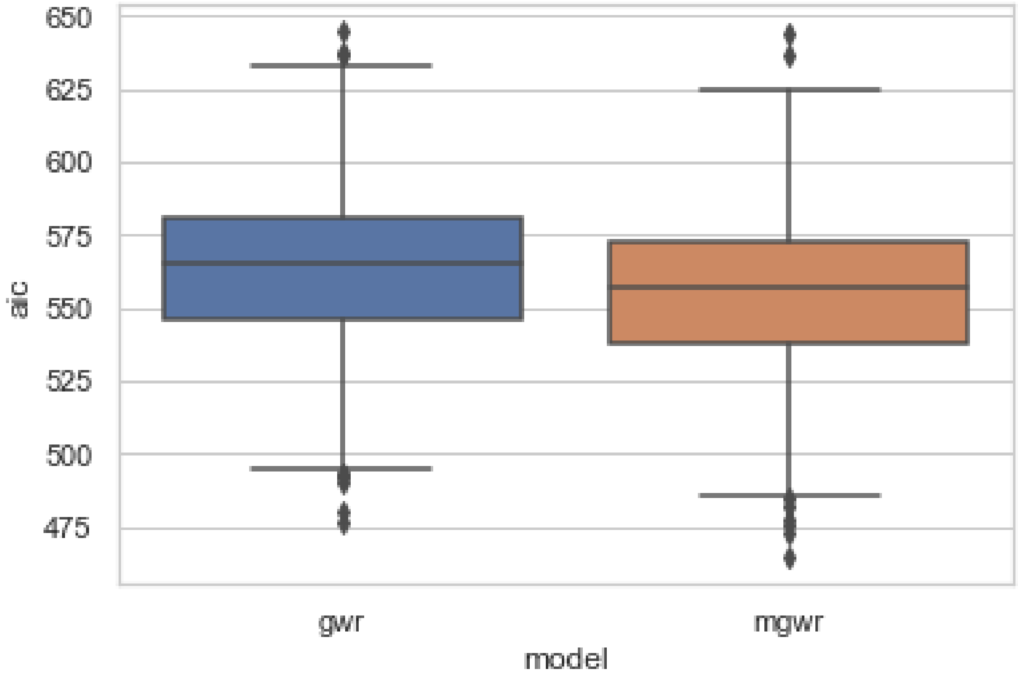


```
In [28]: np.mean(mgwr_aicc), np.mean(gwr_aicc)

Out[28]: (555.2930310559196, 565.2171303944737)

In [29]: np.mean(mgwr_aic), np.mean(gwr_aic)

Out[29]: (555.2543525109128, 563.542676896326)

In [30]: np.mean(mgwr_bic), np.mean(gwr_bic)

Out[30]: (568.5736137906213, 657.6378243512686)
```

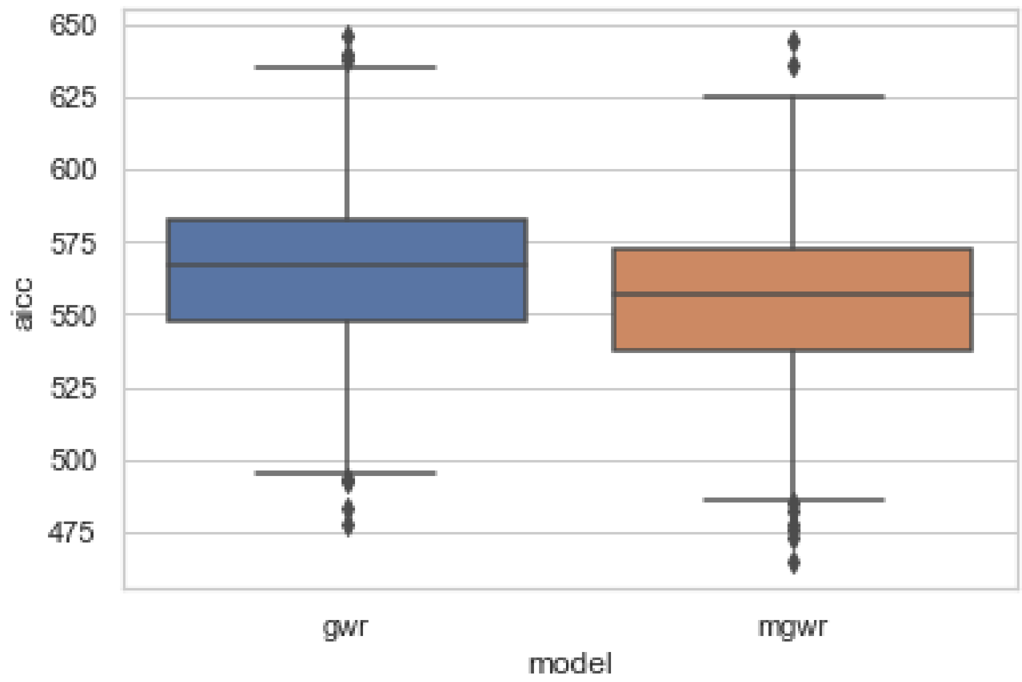
AIC, AICc, BIC Boxplots for comparison

```
In [32]: model=[]
         model = ['gwr']*1000
         model2 = ['mgwr']*1000
         model=model+model2

In [33]: aic=[]
         aic=gwr_aic
         aic=aic+mgwr_aic

In [34]: aicc=[]
         aicc=gwr_aicc
         aicc=aicc+mgwr_aicc
```

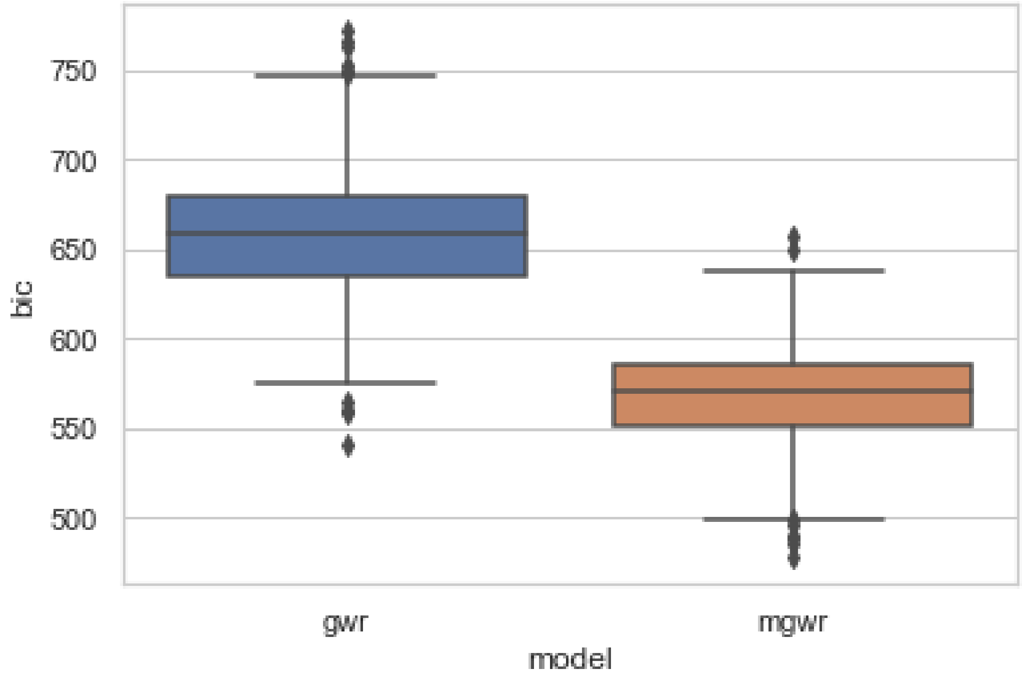
```
In [35]: bic=[]
         bic=gwr_bic
         bic=bic+mgwr_bic

In [36]: d = {'aic':aic,'bic':bic,'aicc':aicc,'model':model}

In [37]: df=pd.DataFrame(data=d)

In [38]: sns.set(style="whitegrid")
         ax = sns.boxplot(y=df['aic'],x=df['model'])
```



```
In [39]: sns.set(style="whitegrid")
         ax = sns.boxplot(y=df['aicc'],x=df['model'])
```



```
In [40]: sns.set(style="whitegrid")
         ax = sns.boxplot(y=df['bic'],x=df['model'])
```



0.0.7 Parameter comparison from MGWR and GWR

```
In [41]: mgwr_params_mean=np.mean(mgwr_params,axis=0)
         gwr_params_mean=np.mean(gwr_params,axis=0)
```

```
In [42]: gwr_params_mean
```

```
Out[42]: array([[0.2846917 , 1.60675938, 0.25596671],
                [0.28371941, 1.60712877, 0.26163037],
                [0.28254898, 1.60802699, 0.26836111],
                ...,
                [0.28030459, 1.7171863 , 0.64758872],
                [0.27994884, 1.68247975, 0.6534747 ],
                [0.27958057, 1.65168921, 0.65852034]])
```

```
In [43]: 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 [44]: 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 [45]: B0_mgwr=B0_mgwr.reshape(25,25)
         B1_mgwr=B1_mgwr.reshape(25,25)
         B2_mgwr=B2_mgwr.reshape(25,25)
```

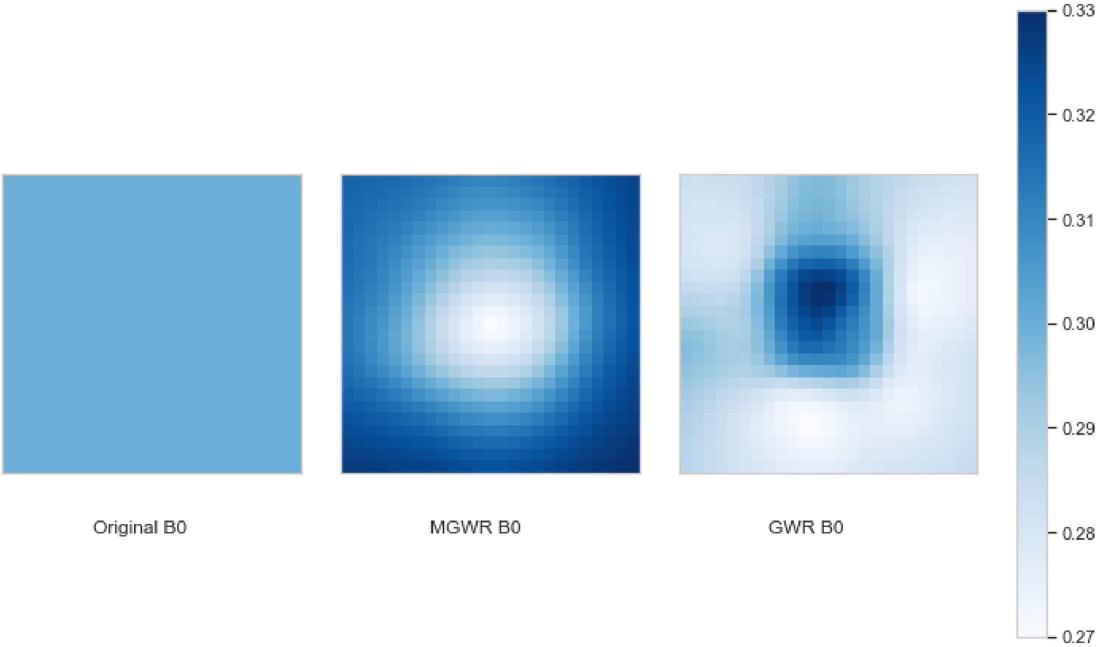
```
In [46]: np.mean(B1_mgwr),np.mean(B1_gwr),np.mean(B1)
```

```
Out[46]: (2.5987313746451686, 2.452124883320957, 1.147104517676957)
```

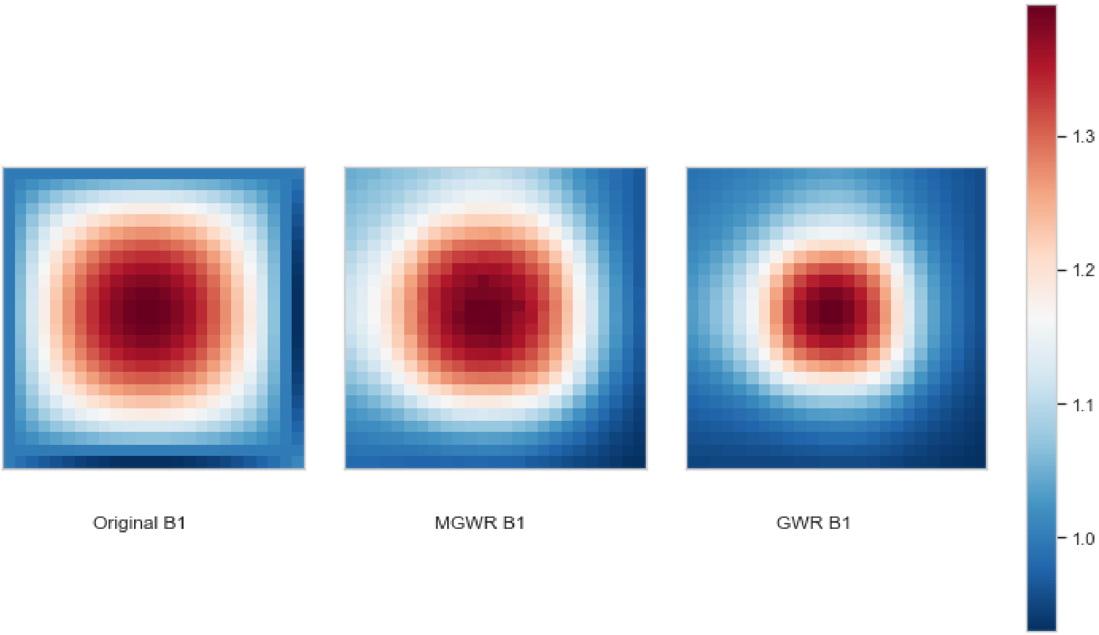
```
In [47]: B0_gwr=B0_gwr.reshape(25,25)
         B1_gwr=B1_gwr.reshape(25,25)
         B2_gwr=B2_gwr.reshape(25,25)
```

```
In [48]: 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()
```

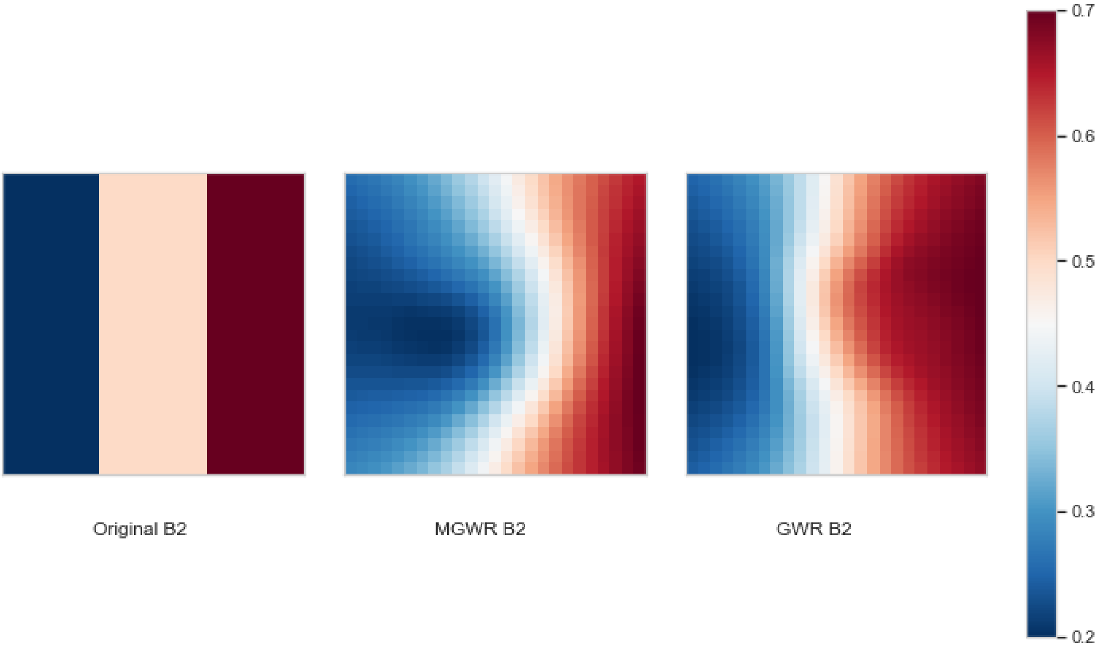


```
In [49]: fig, (ax, ax2,ax3, cax) = plt.subplots(ncols=4,figsize=(10,6),
                           gridspec_kw={"width_ratios":[1,1,1, 0.1],"height_ratios":[1]})
         fig.subplots_adjust(wspace=0.3)
         im = ax.imshow(B1, extent=[0,10, 0, 10], origin='lower',cmap='RdBu_r')
         ax.text(3, -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 [50]: 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 [51]: 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 [52]: 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]
```

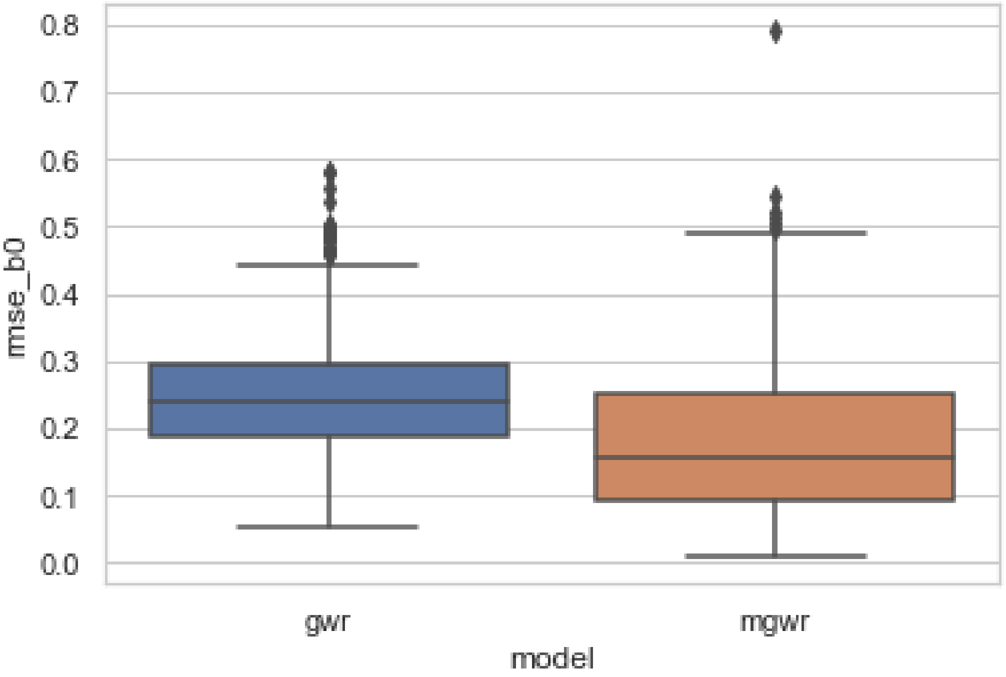
```
In [53]: b0 = B0.reshape(-1,1)
         b1 = B1.reshape(-1,1)
         b2 = B2.reshape(-1,1)
```

0.0.9 B_0

```
In [55]: rmse_b0_m=[]
         for i in range(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)
```

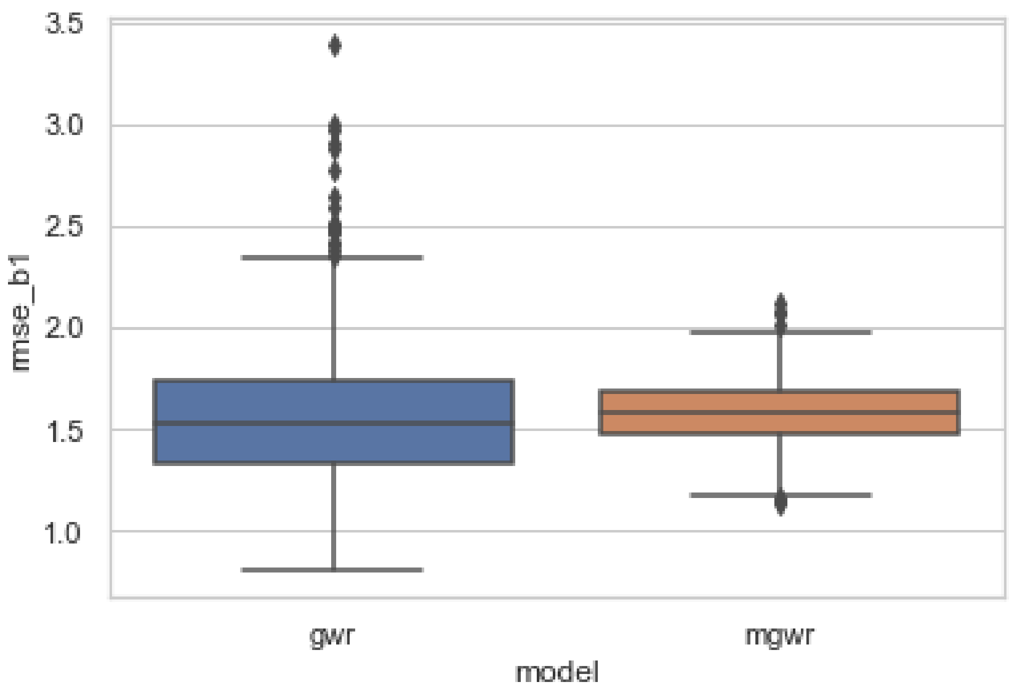
```
In [56]: model=[]
         model = ['gwr']*1000
         model2 = ['mgwr']*1000
         model=model+model2

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

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

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



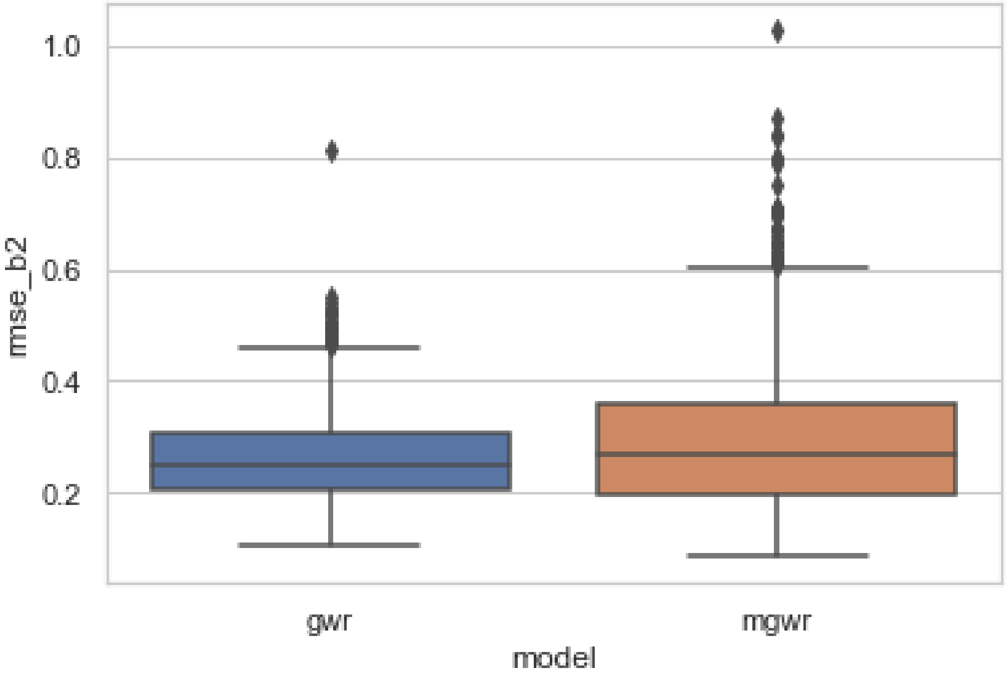
```
In [62]: np.sqrt((np.sum((b1 - (np.hsplit(mgwr_params_mean,3)[1]))**2))/625)

Out[62]: 1.5218558005607021

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

Out[63]: 1.487375751104697
```

0.0.11 B_2

```
In [65]: 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 [66]: 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 [67]: sns.set(style="whitegrid")
         ax = sns.boxplot(y=df['rmse_b2'],x=df['model'])
```



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

```
Out[68]: 0.16342922217077274
```

```
In [69]: np.sqrt((np.sum((b2 - (np.hsplit(gwr_params_mean,3)[2]))**2))/625)
```

```
Out[69]: 0.07374405903986679
```