aede

April 1, 2019

1 Análise espacial

In [ ]:
```
%matplotlib inline
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
import seaborn as sns
import geopandas as gp
import pysal as ps
import palettable
import nupis
import mapclassify as mc
import matplotlib.lines as mlines
import pysal.contrib.viz.mapping as maps
import pickle
from plotar import plotar
plt.style.use('seaborn-whitegrid')

plt.rc('figure', max_open_warning = 50)
pd.options.display.max_rows = 1000

# dicionário com os dataframes dos anos
temp = open('../dados/separado_rgi.pickle', 'rb')
separado_rgi = pickle.load(temp)

# informações sobre MG - geo e dados
mg = pd.read_csv('../dados/mg-mapas.csv', 'encoding=utf-8')
mg = gp.GeoDataFrame(mg)   # transforma em geopandas
```

```
C:\Users\Renan\Anaconda3\lib\site-packages\ipykernel_launcher.py:25: ParserWarning: Falling back
```

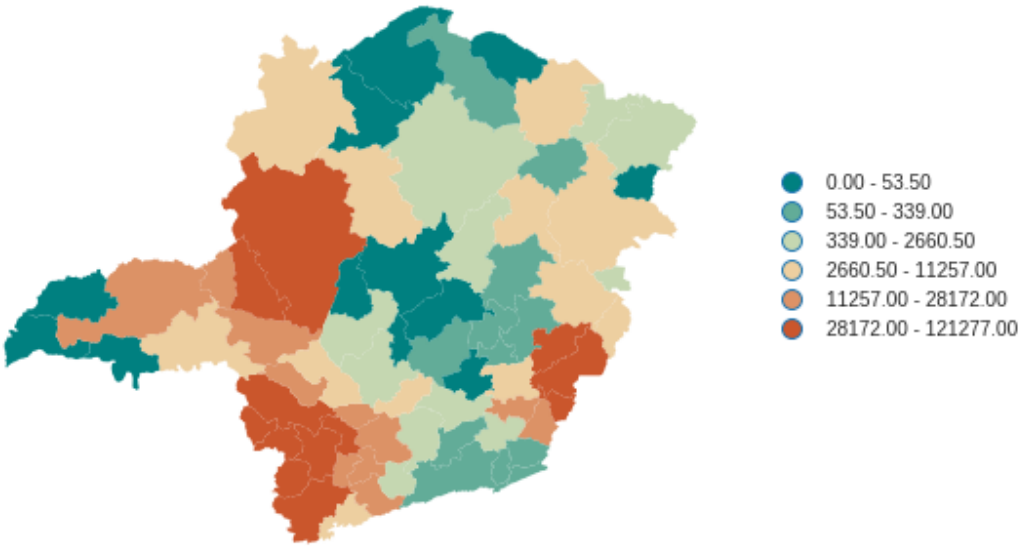
In [3]:
```
# informações para a legenda dos mapas temáticos
referencia = [500, 1000, 1500, 2000, 2500, 5000]
labels = ['0 - 500', '501 - 1000', '1001 - 1500', '1501 - 2000', '2001 - 2500', '2501 -
paleta = palettable.cartocolors.diverging.Geyser_3.mpl_colormap

anos = mg.ano.unique()
```

```
mark1 = mlines.Line2D([0], [0], marker='o', color='w', markerfacecolor='#0A7D7D', marker
mark2 = mlines.Line2D([0], [0], marker='o', color='w', markerfacecolor='#4CA795', marker
mark3 = mlines.Line2D([0], [0], marker='o', color='w', markerfacecolor='#9FC1A8', marker
mark4 = mlines.Line2D([0], [0], marker='o', color='w', markerfacecolor='#F8F0C2', marker
mark5 = mlines.Line2D([0], [0], marker='o', color='w', markerfacecolor='#F3BB80', marker
mark6 = mlines.Line2D([0], [0], marker='o', color='w', markerfacecolor='#DD8150', marker
handles = [mark1, mark2, mark3, mark4, mark5, mark6]
```

In [4]:
```
separado_rgi[2017].shape
```

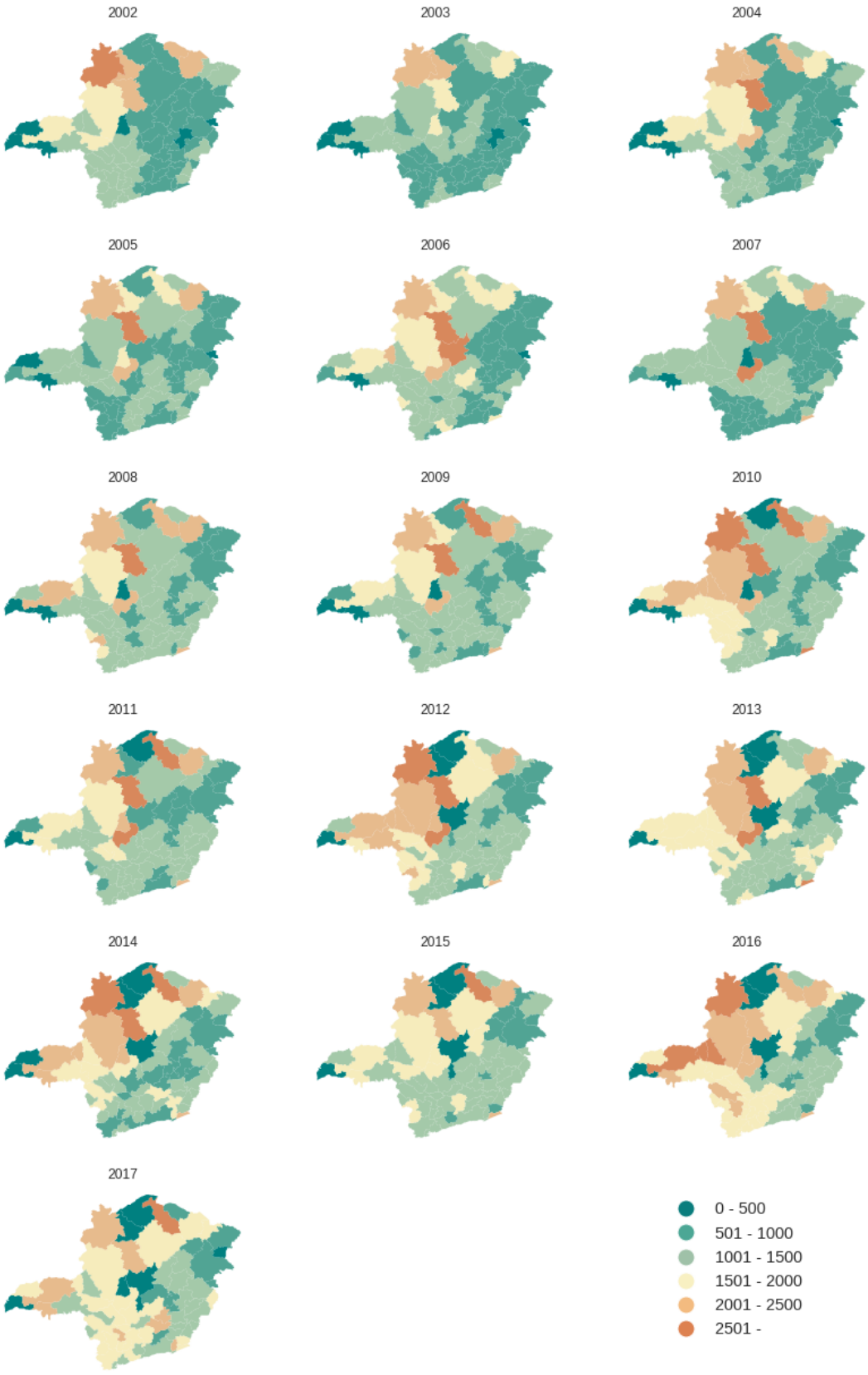
Out[4]: (70, 10)

In [4]:
```
# intervalos iguais
ax = separado_rgi[2017].plot(column='area_colhida', scheme='quantiles', k=6,
             linewidth=0, figsize=(7,7), legend=True, cmap=paleta)
ax.set_axis_off()
leg = ax.get_legend()
leg.set_bbox_to_anchor((1.4, 0.7));
```



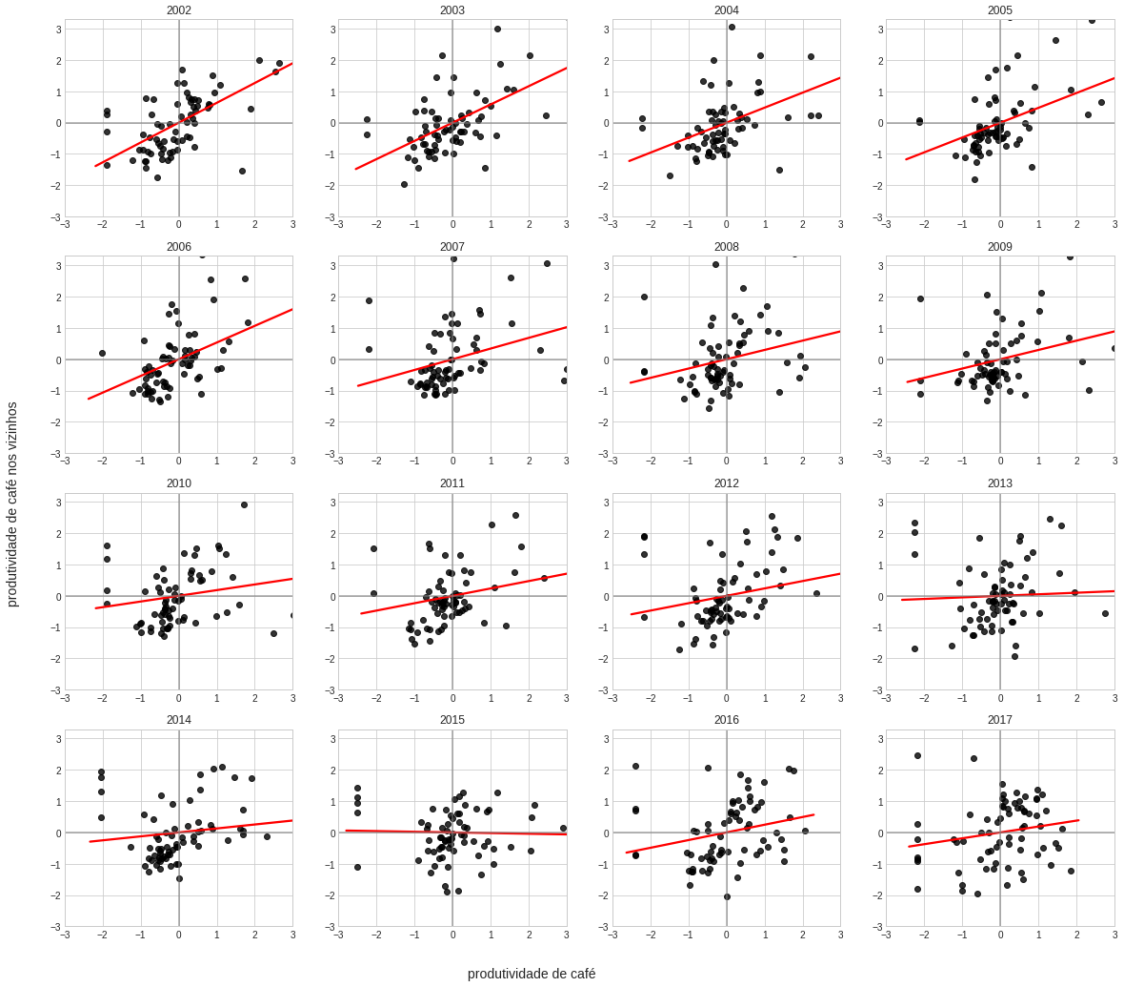
In [21]:
```
# mapas temáticos com quantidade desigual
fig, axes = plt.subplots(6, 3, figsize=(14, 22))
for i, ax in enumerate(axes.reshape(-1)):
    if i < len(anos):
        valores = separado_rgi[anos[i]]['rendimento'].values.reshape(separado_rgi[anos[
        quantis = mc.User_Defined(valores, referencia)
        cl = [labels[i] for i in quantis.yb]
        df = separado_rgi[anos[i]].assign(cl=cl)
```

```
            df.cl = pd.Categorical(df.cl, ordered=True, categories=labels)
            ax = plotar(df, column='cl', categorical=True, linewidth=0, figsize=(5,5), lege
                    cmap=paleta, labels=labels, ax=ax)

            ax.set_title(anos[i])

        ax.set_axis_off()

fig.legend(handles=handles, labels=labels, loc='lower right', bbox_to_anchor=(0.75, 0.0
```


2002
2003
2004
2005
2006
2007
2008
2009
2010
2011
2012
2013
2014
2015
2016
2017
0 - 500
501 - 1000
1001 - 1500
1501 - 2000
2001 - 2500
2501 -

```
In [36]: separado_rgi[2017].query('2500>rendimento>2000').shape

Out[36]: (8, 13)

In [5]: # MATRIZ DO TIPO RAINHA

        # dicionário com os resultados por ano
        resultados_rgi = {}
        w = ps.queen_from_shapefile('../dados/mg_rgi_novo.shp')
        for i in range(len(anos)):
            IM = ps.Moran(separado_rgi[anos[i]].rendimento, w)
            resultados_rgi[anos[i]] = [round(IM.I, 3), IM.p_sim]
            print(f'{anos[i]}: I = {round(IM.I, 3)} valor-p = {IM.p_sim}')

2002: I = 0.347 valor-p = 0.001
2003: I = 0.371 valor-p = 0.001
2004: I = 0.306 valor-p = 0.001
2005: I = 0.309 valor-p = 0.001
2006: I = 0.37 valor-p = 0.001
2007: I = 0.202 valor-p = 0.009
2008: I = 0.167 valor-p = 0.012
2009: I = 0.181 valor-p = 0.009
2010: I = 0.099 valor-p = 0.057
2011: I = 0.134 valor-p = 0.026
2012: I = 0.12 valor-p = 0.051
2013: I = 0.022 valor-p = 0.284
2014: I = 0.064 valor-p = 0.141
2015: I = -0.01 valor-p = 0.475
2016: I = 0.13 valor-p = 0.038
2017: I = 0.09 valor-p = 0.083


In [6]: quadrantes_anos = {}    # dicionário que conterá as listas de rgi por quadrante para os
                                # ordem: BB, BA, AA, AB
        n = len(mg.ano.unique())
        anos = mg.ano.unique()

        fig, axes = plt.subplots(4, 4, figsize=(17, 15))
        for i, ax in enumerate(axes.flat, 0):
            separado_rgi[anos[i]] = separado_rgi[anos[i]].reset_index()  # fazer com que os índi
            IM = ps.Moran(separado_rgi[anos[i]].rendimento, w)
            resultados_rgi[anos[i]] = [round(IM.I, 3), IM.p_sim]
            y_norm = (IM.y - IM.y.mean()) / IM.y.std()
            y_lag = ps.lag_spatial(IM.w, IM.y)
            y_lag_norm = (y_lag - y_lag.mean()) / y_lag.std()
            dados = pd.DataFrame({'y':IM.y, 'y_norm':y_norm,
```

```
                        'y_lag':y_lag, 'y_lag_norm':y_lag_norm})
    ax = sns.regplot('y_norm', 'y_lag_norm', data=dados, ci=None,
                color='black', line_kws={'color':'red'}, ax=ax)
    ax.axvline(0, c='gray', alpha=0.7)
    ax.axhline(0, c='gray', alpha=0.7)

#     limits = np.array([y_norm.min(), y_norm.max(), y_lag_norm.min(), y_lag_norm.max()]
#     limits = np.abs(limits).max()
    border = 0.02
    ax.set_xlim(-3, 3)     # fixando os limites
    ax.set_ylim(-3, 3.3)   # fixando os limites
    title = f'{anos[i]}'
    ax.set_xlabel('')
    ax.set_ylabel('')
    ax.set_title(title)

    # guardar as rgi de cada quadrante de cada ano
    lista = []
    BB = dados.query('y_norm < 0 & y_lag_norm < 0')
    BA = dados.query('y_norm < 0 & y_lag_norm > 0')
    AA = dados.query('y_norm > 0 & y_lag_norm > 0')
    AB = dados.query('y_norm > 0 & y_lag_norm < 0')
    BB_rgi = separado_rgi[anos[i]].nome_rgi[BB.index].values.tolist()
    BA_rgi = separado_rgi[anos[i]].nome_rgi[BA.index].values.tolist()
    AA_rgi = separado_rgi[anos[i]].nome_rgi[AA.index].values.tolist()
    AB_rgi = separado_rgi[anos[i]].nome_rgi[AB.index].values.tolist()
    lista.append(BB_rgi)
    lista.append(BA_rgi)
    lista.append(AA_rgi)
    lista.append(AB_rgi)
    quadrantes_anos[anos[i]] = lista

plt.gcf().text(0.4, -0.05, 'produtividade de café', ha='center', fontsize=14)
plt.gcf().text(-0.05, 0.5, 'produtividade de café nos vizinhos', rotation=90, fontsize=1
plt.subplots_adjust(left=-0.0005, bottom=-0.0005);
```



para ver as regiões em cada quadrante: quadrantes_anos['ano'][i]
i=
0-BAIXO-BAIXO
1-BAIXO-ALTO
2-ALTO-ALTO
3-ALTO-BAIXO

```
In [119]: #Verificar o comportamento de cada região isoladamente
          for i in range(2002,2018):
              for j in range(0,4):
                  if quadrantes_anos[i][j].count('Três Pontas - Boa Esperança') >0:
                      print(i,j)
```

```
2002 2
2003 0
2004 1
2005 0
2006 3
```

```
2007 0
2008 0
2009 3
2010 3
2011 3
2012 3
2013 2
2014 0
2015 0
2016 2
2017 2
```

```
In [14]: quadrantes_anos[2017][2]
```

```
Out[14]: ['Salinas',
          'Cataguases',
          'São João Nepomuceno - Bicas',
          'Além Paraíba',
          'Conselheiro Lafaiete',
          'São João del Rei',
          'Varginha',
          'Passos',
          'Alfenas',
          'Lavras',
          'Guaxupé',
          'Três Corações',
          'Três Pontas - Boa Esperança',
          'São Sebastião do Paraíso',
          'Campo Belo',
          'Piumhi',
          'Pouso Alegre',
          'Poços de Caldas',
          'São Lourenço',
          'Caxambu - Baependi',
          'Uberaba',
          'Araxá',
          'Uberlândia',
          'Monte Carmelo',
          'Patrocínio',
          'Formiga',
          'Oliveira']
```

```
In [122]: #número de regiões por quadrante ao lonog do tempo
          for i in range (2002,2018):
              print( i, len(quadrantes_anos[i][0]) )
```

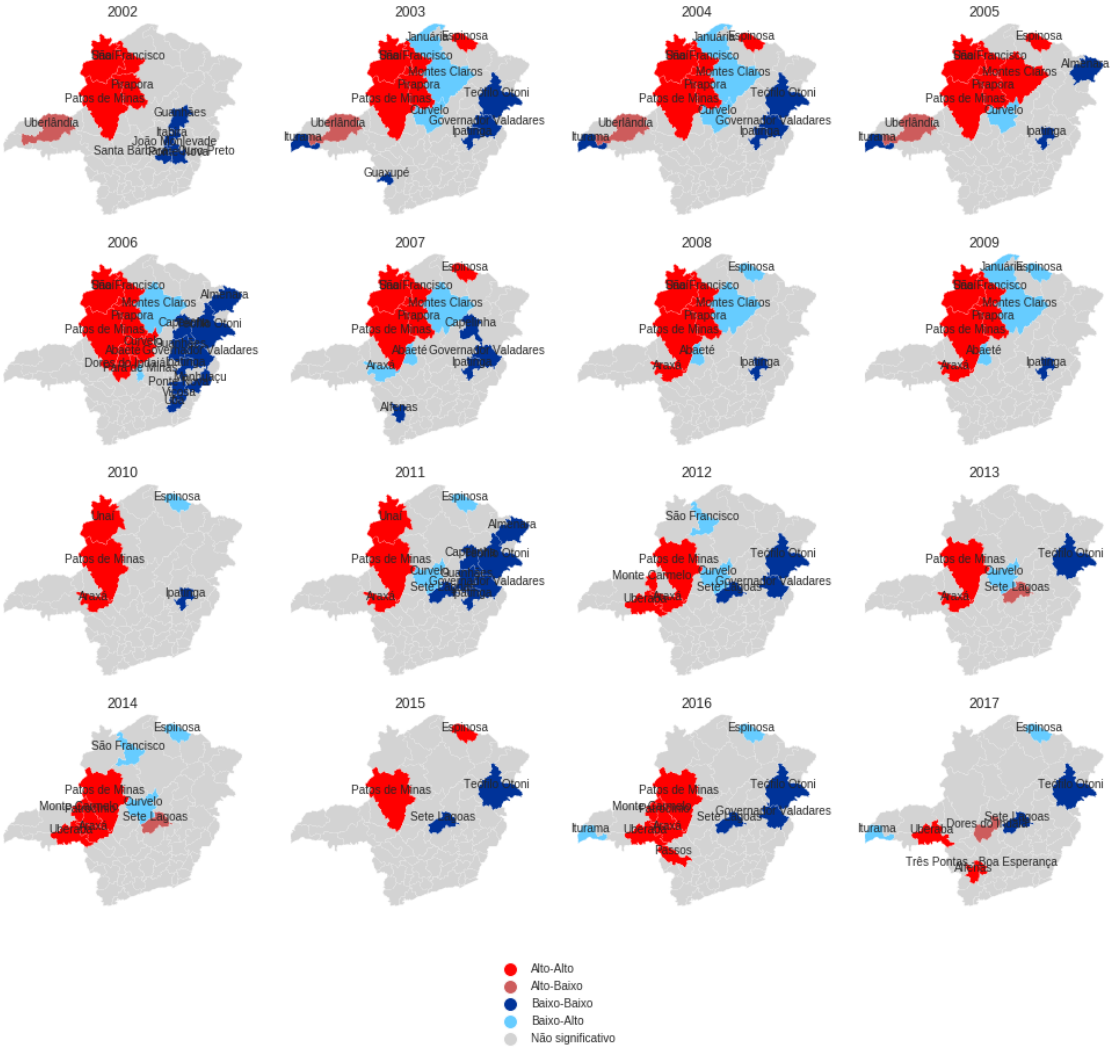
```
2002 31
2003 28
```

```
2004 34
2005 35
2006 29
2007 34
2008 32
2009 35
2010 31
2011 30
2012 32
2013 23
2014 34
2015 25
2016 24
2017 21
```

```
In [17]: ALTO_ALTO = mlines.Line2D([0], [0], marker='o', color='w',  markerfacecolor='#FF0505',
         ALTO_BAIXO = mlines.Line2D([0], [0], marker='o', color='w', markerfacecolor='#CD5C5C',
         BAIXO_BAIXO = mlines.Line2D([0], [0], marker='o', color='w', markerfacecolor='#003399',
         BAIXO_ALTO = mlines.Line2D([0], [0], marker='o', color='w', markerfacecolor='#66CCFF',
         NAO_SIG = mlines.Line2D([0], [0], marker='o', color='w', markerfacecolor='#D3D3D3', mar


         legenda=['Alto-Alto', 'Alto-Baixo', 'Baixo-Baixo', 'Baixo-Alto', 'Não significativo']
         handles = [ALTO_ALTO, ALTO_BAIXO, BAIXO_BAIXO, BAIXO_ALTO, NAO_SIG]
```

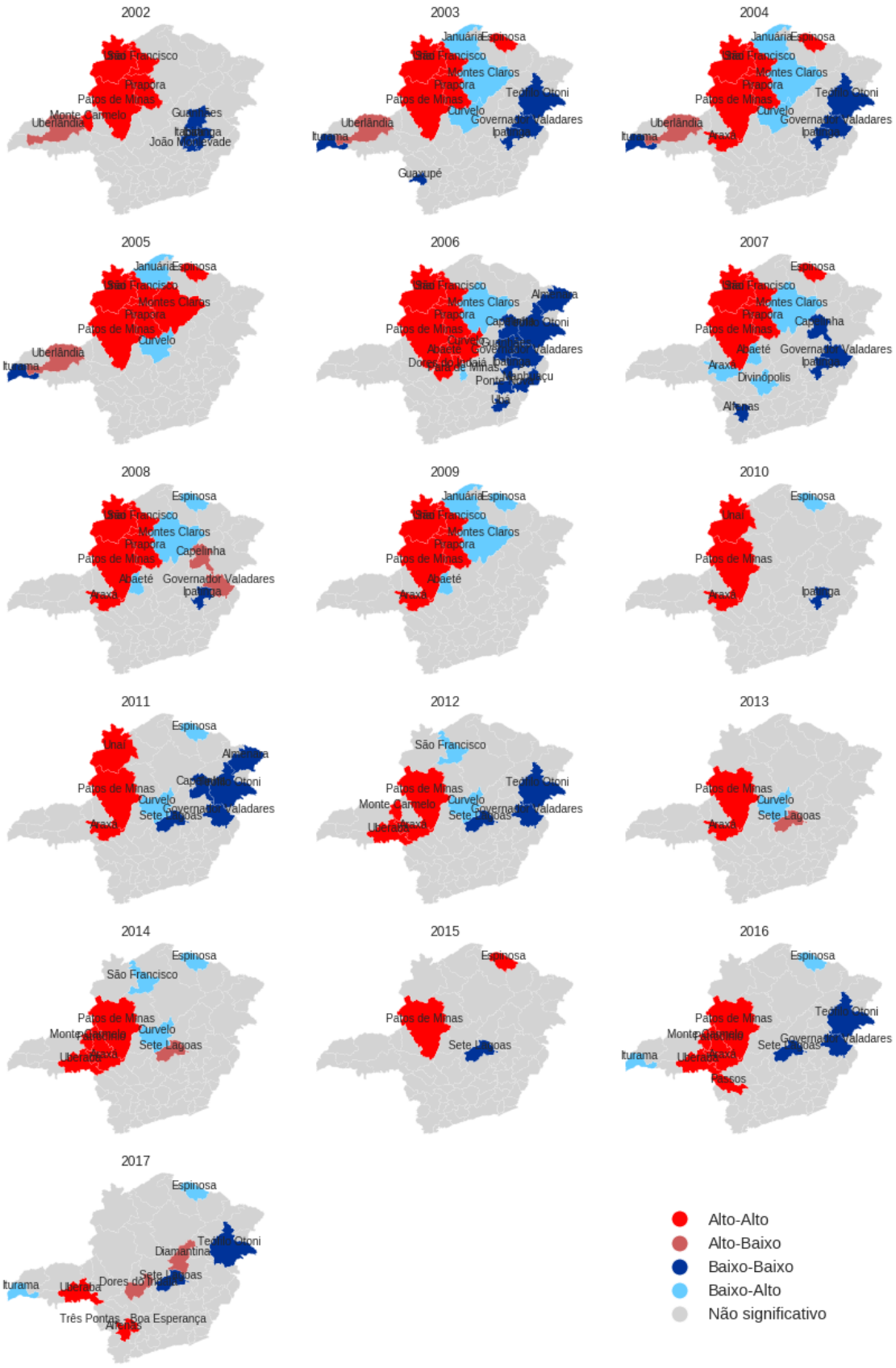
```
In [8]: # mapas LISA por ano com apenas uma legenda
        import pysal.contrib.viz.mapping as maps
        fig, axes = plt.subplots(4, 4, figsize=(17, 14))
        for i, ax in enumerate(axes.flat, 0):
            lisa = ps.Moran_Local(separado_rgi[anos[i]].rendimento, w)
            shp = ps.open('../dados/mg_rgi_novo.shp')
            base = maps.map_poly_shp(shp)
            base = maps.base_lisa_cluster(base, lisa, p_thres=0.05)
            base.set_edgecolor('1')
            base.set_linewidth(0.1)
            ax = maps.setup_ax([base], [shp.bbox], ax=ax)
            ax.set_title(anos[i])
            boxes, labels = maps.lisa_legend_components(lisa, p_thres=0.05)
            # identificar rgis
            sig = lisa.p_sim < 0.05  # identificar significativos
            posicoes = np.where(sig)
            df_ano = separado_rgi[anos[i]].reset_index()
            rgi_escolhidas = df_ano.iloc[posicoes[0]]
            for j in rgi_escolhidas.index:
                ax.text(rgi_escolhidas.geometry.centroid[j].coords[0][0], rgi_escolhidas.geometr
                     rgi_escolhidas.nome_rgi[j],
                     fontsize=10, horizontalalignment='center', verticalalignment='bottom');
```

```
fig.legend(handles=handles,labels=legenda, loc='lower center');
# plt.savefig('imagens/lisa_rgi.png', bbox_inches='tight')
```



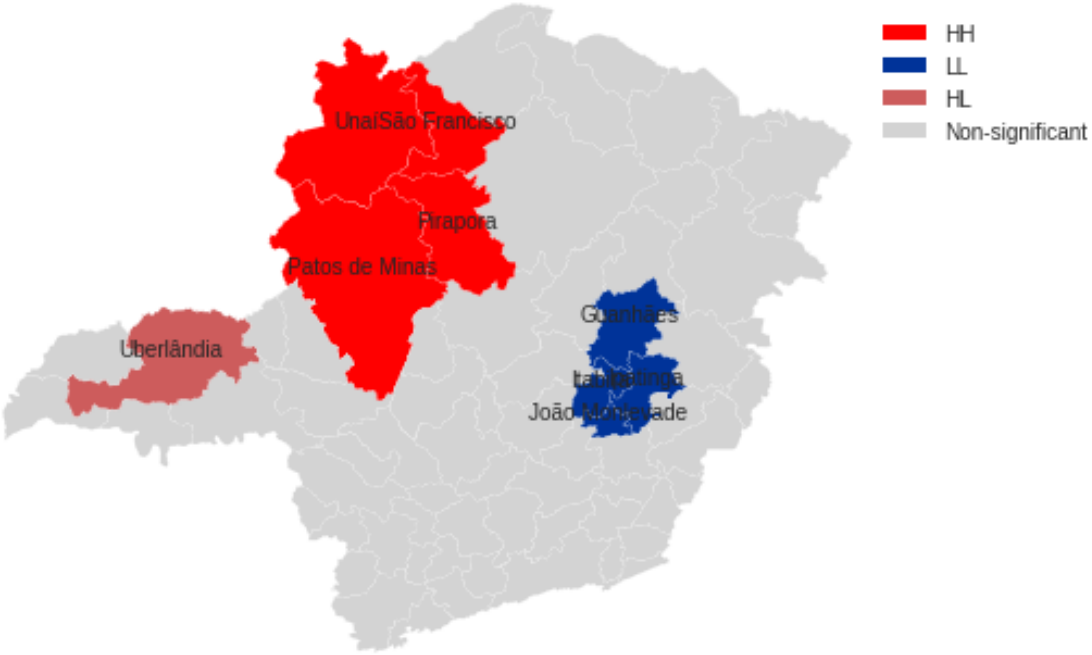
In [18]:
```
# mapas temáticos com quantidade desigual
fig, axes = plt.subplots(6, 3, figsize=(14, 22))
for i, ax in enumerate(axes.reshape(-1)):
    if i < len(anos):
        lisa = ps.Moran_Local(separado_rgi[anos[i]].rendimento, w)
        shp = ps.open('../dados/mg_rgi_novo.shp')
        base = maps.map_poly_shp(shp)
        base = maps.base_lisa_cluster(base, lisa, p_thres=0.05)
        base.set_edgecolor('1')
        base.set_linewidth(0.1)
        ax = maps.setup_ax([base], [shp.bbox], ax=ax)
```

```
        ax.set_title(anos[i])
        boxes, labels = maps.lisa_legend_components(lisa, p_thres=0.05)
        # identificar rgis
        sig = lisa.p_sim < 0.05  # identificar significativos
        posicoes = np.where(sig)
        df_ano = separado_rgi[anos[i]].reset_index()
        rgi_escolhidas = df_ano.iloc[posicoes[0]]
        for j in rgi_escolhidas.index:
            ax.text(rgi_escolhidas.geometry.centroid[j].coords[0][0], rgi_escolhidas.ge
                rgi_escolhidas.nome_rgi[j],
                fontsize=10, horizontalalignment='center', verticalalignment='bottom')
    ax.set_axis_off()

fig.legend(handles=handles, labels=legenda, loc='lower right', bbox_to_anchor=(0.75, 0.
```

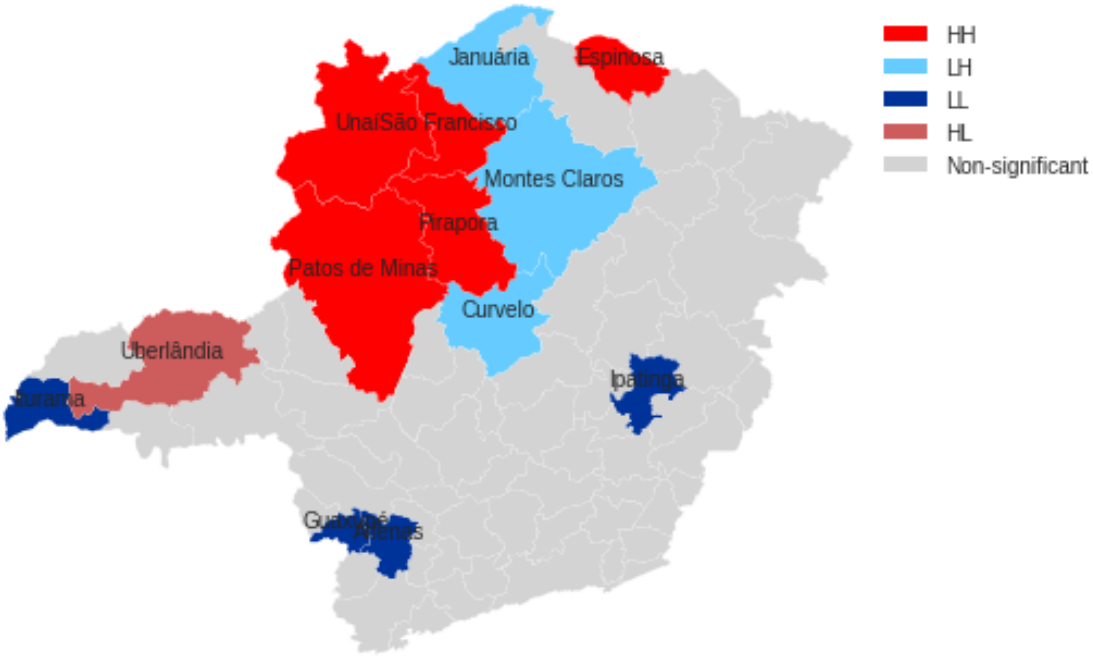

2002
2003
2004
2005
2006
2007
2008
2009
2010
2011
2012
2013
2014
2015
2016
2017
Alto-Alto
Alto-Baixo
Baixo-Baixo
Baixo-Alto
Não significativo

In [10]:
```
# um mapa por linha
import pysal.contrib.viz.mapping as maps
for i in range(len(anos)):
    lisa = ps.Moran_Local(separado_rgi[anos[i]].rendimento, w)
    fig = plt.figure(figsize=(7, 5.5))
    fig.suptitle(anos[i], fontsize=12)
    shp = ps.open('../dados/mg_rgi_novo.shp')
    base = maps.map_poly_shp(shp)
    base = maps.base_lisa_cluster(base, lisa, p_thres=0.05)
    base.set_edgecolor('1')
    base.set_linewidth(0.1)
    ax = maps.setup_ax([base], [shp.bbox])

    boxes, labels = maps.lisa_legend_components(lisa, p_thres=0.05)
    plt.legend(boxes, labels, fancybox=True, bbox_to_anchor=(1.3, 1));

    # identificar rgis
    sig = lisa.p_sim < 0.05  # identificar significativos
    posicoes = np.where(sig)
    df_ano = separado_rgi[anos[i]].reset_index()
    df_ano['quad'] = lisa.q
    rgi_escolhidas = df_ano.iloc[posicoes[0]]
    for j in rgi_escolhidas.index:
        plt.text(rgi_escolhidas.geometry.centroid[j].coords[0][0], rgi_escolhidas.geome
             rgi_escolhidas.nome_rgi[j],
             fontsize=10, horizontalalignment='center', verticalalignment='bottom');
plt.show();
```

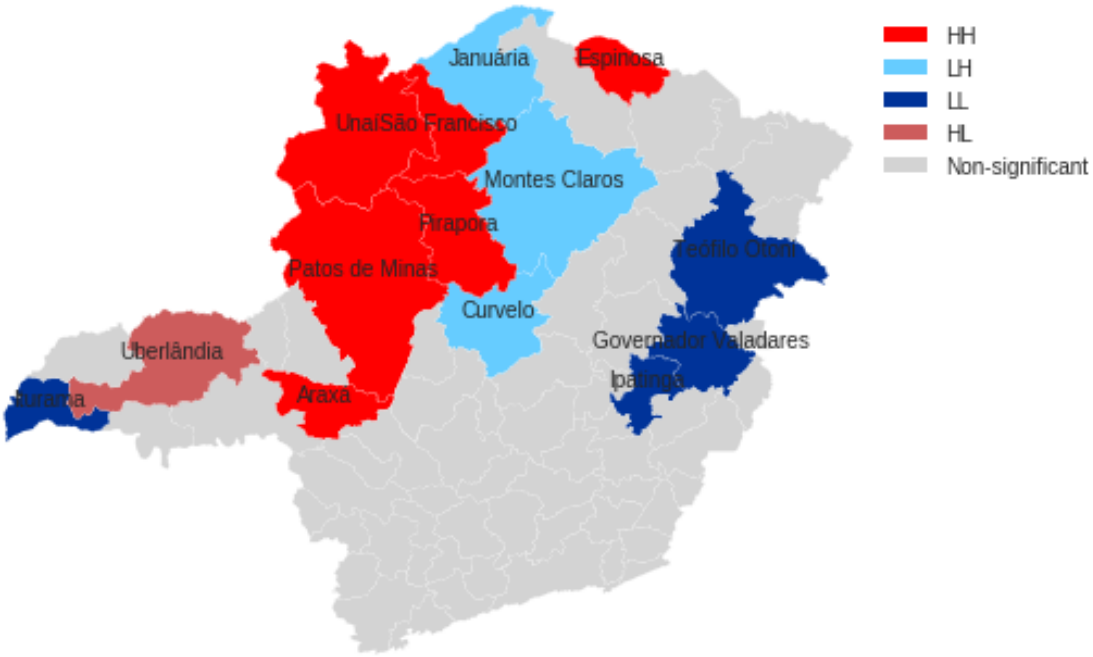
2002


HH
LL
HL
Non-significant
Unaí/São Francisco
Pirapora
Patos de Minas
Uberlândia
Guanhães
João Monlevade

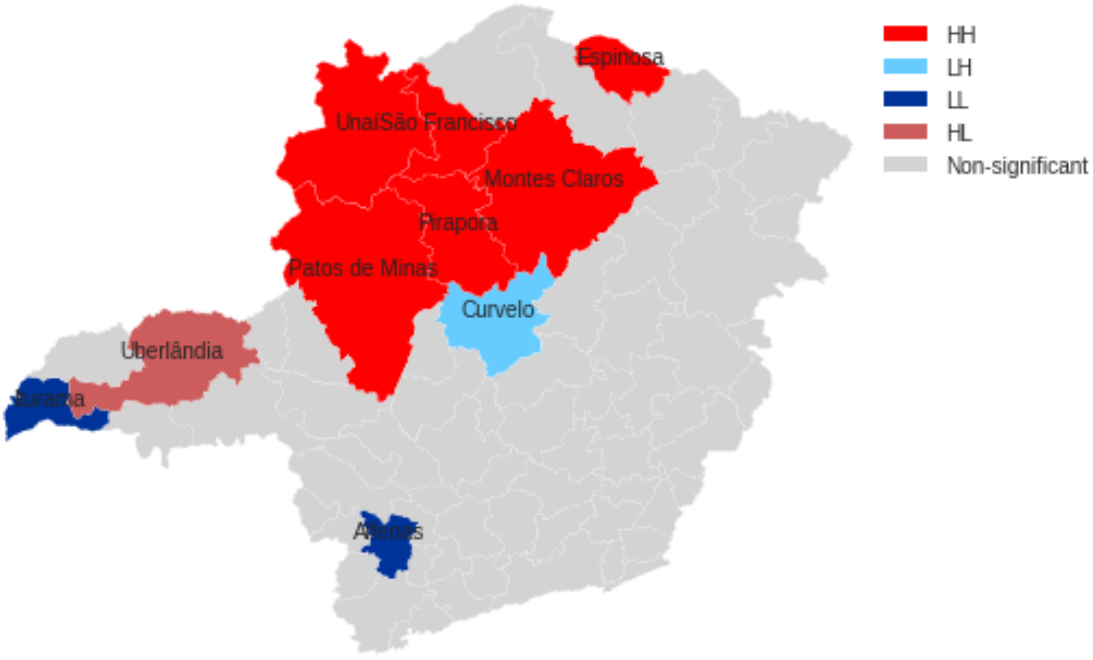
2003


HH
LH
LL
HL
Non-significant
Januária
Espinosa
Unaí/São Francisco
Montes Claros
Pirapora
Patos de Minas
Curvelo
Uberlândia
Ipatinga

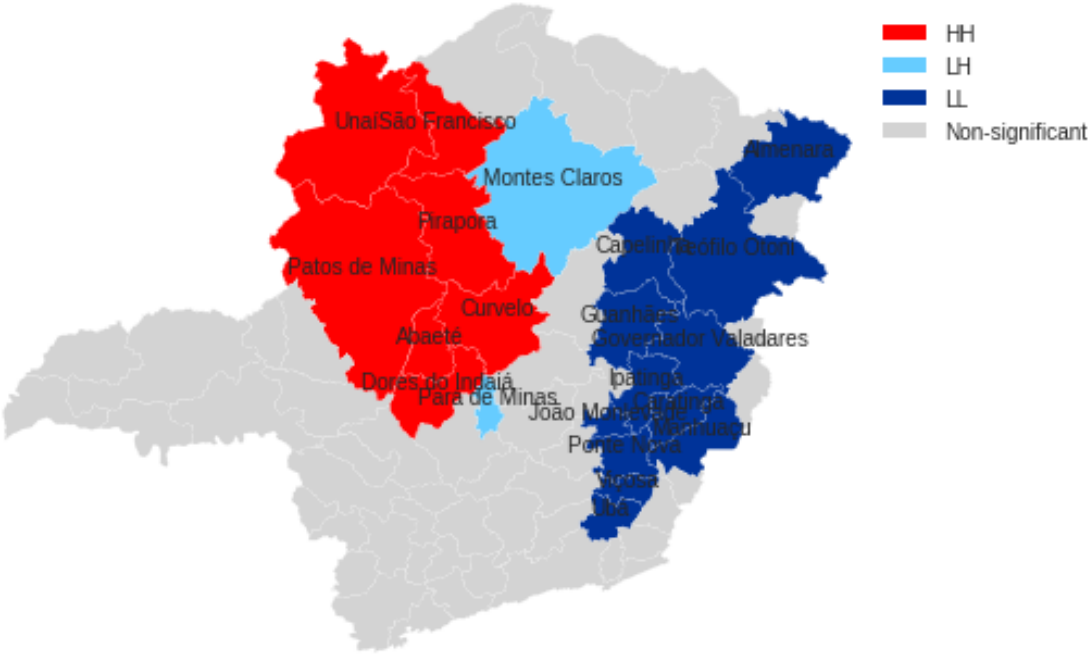
2004


HH
LH
LL
HL
Non-significant
Januária
Espinosa
Unaí/São Francisco
Montes Claros
Pirapora
Patos de Minas
Teófilo Otoni
Curvelo
Governador Valadares
Ipatinga
Uberlândia
Araxá
Juruaia

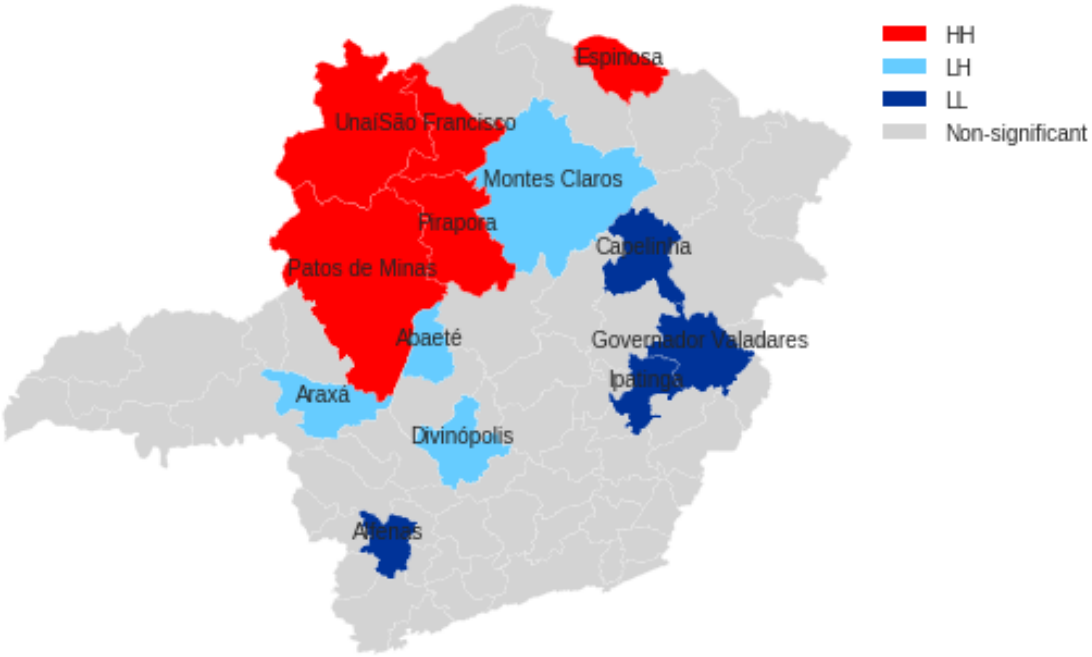
2005


HH
LH
LL
HL
Non-significant
Espinosa
Unaí/São Francisco
Montes Claros
Pirapora
Patos de Minas
Curvelo
Uberlândia
Juruaia
Alfenas

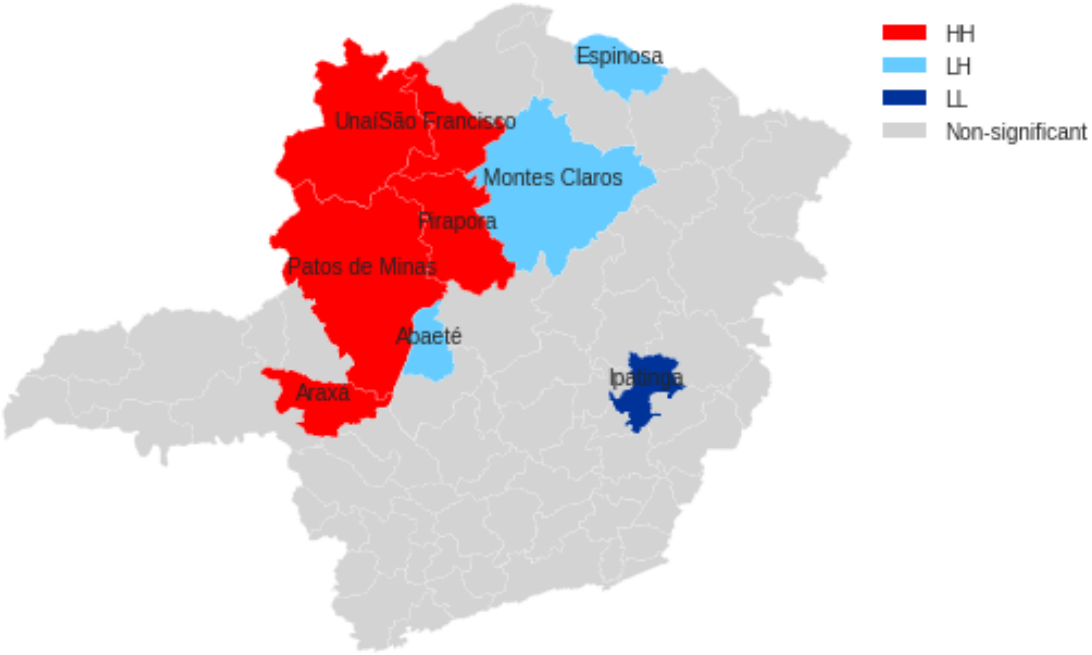
2006


HH
LH
LL
Non-significant
Unaí/São Francisco
Montes Claros
Pirapora
Patos de Minas
Curvelo
Abaeté
Dores do Indaiá
Pará de Minas
Almenara
Governador Valadares
Ipatinga
Ponte Nova

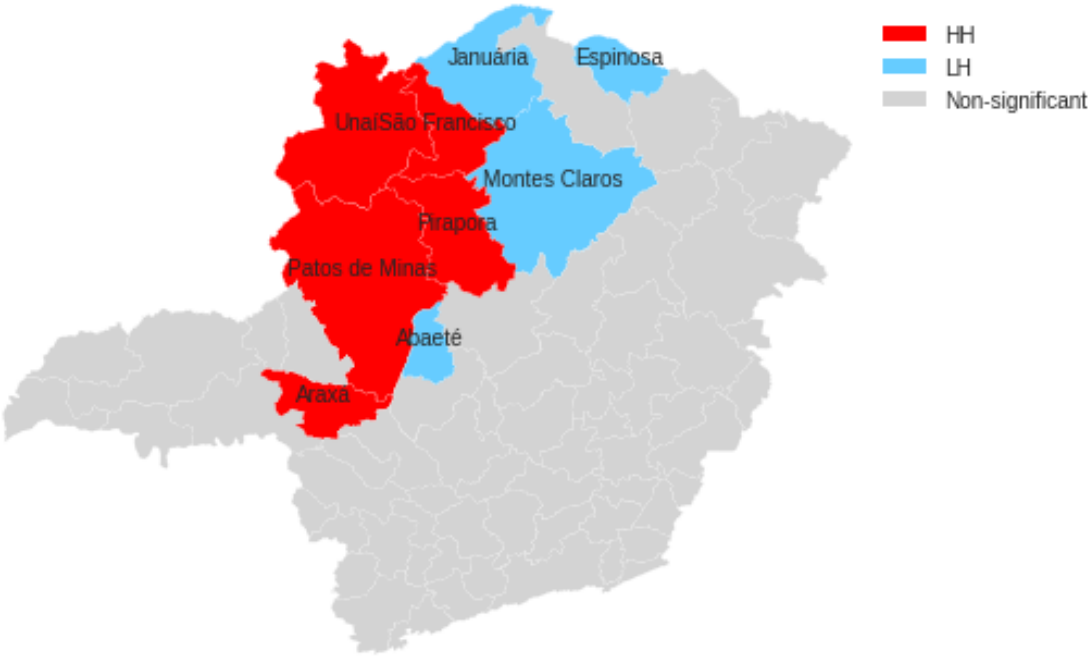
2007


HH
LH
LL
Non-significant
Espinosa
Unaí/São Francisco
Montes Claros
Pirapora
Patos de Minas
Abaeté
Araxá
Divinópolis
Governador Valadares
Ipatinga

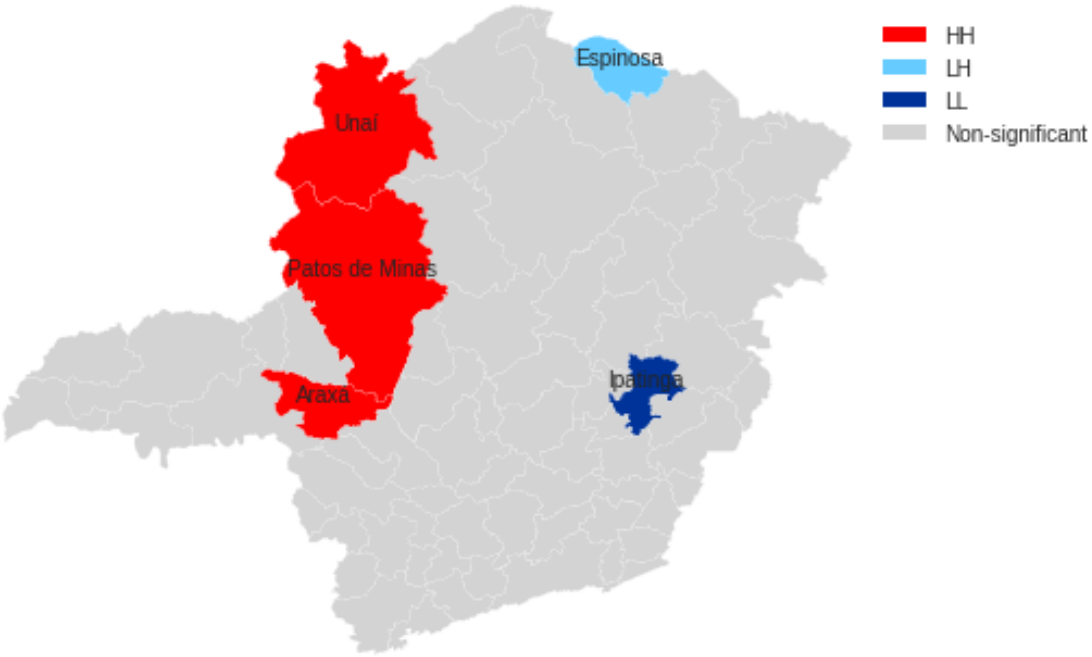
2008


HH
LH
LL
Non-significant
Espinosa
Una/São Francisco
Montes Claros
Pirapora
Patos de Minas
Abaeté
Araxá
Ipatinga

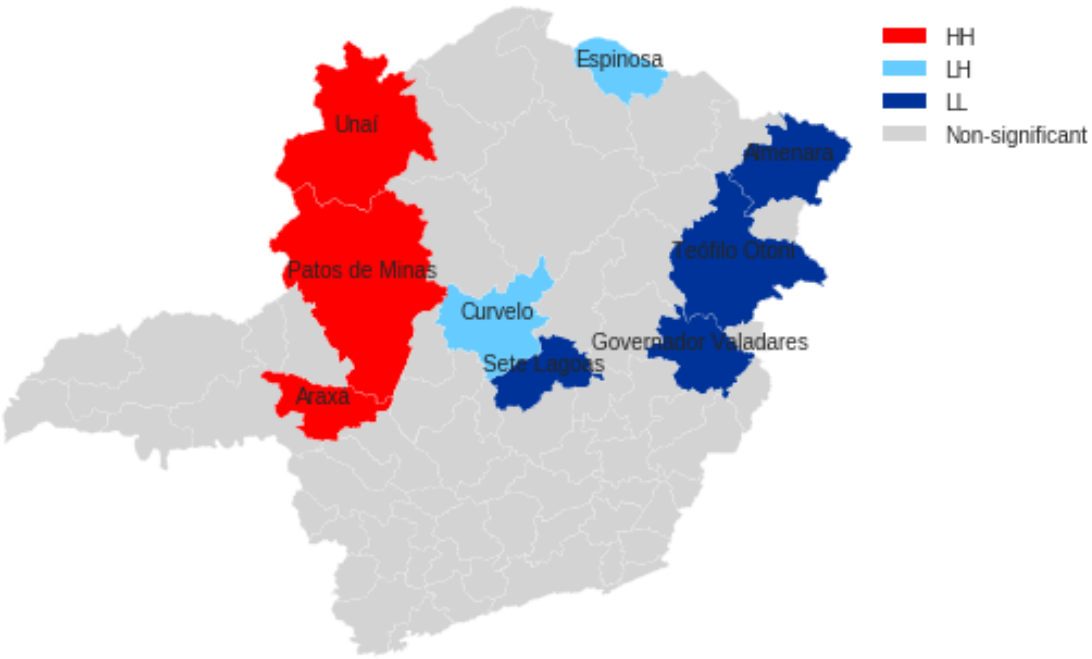
2009


HH
LH
Non-significant
Januária
Espinosa
Una/São Francisco
Montes Claros
Pirapora
Patos de Minas
Abaeté
Araxá

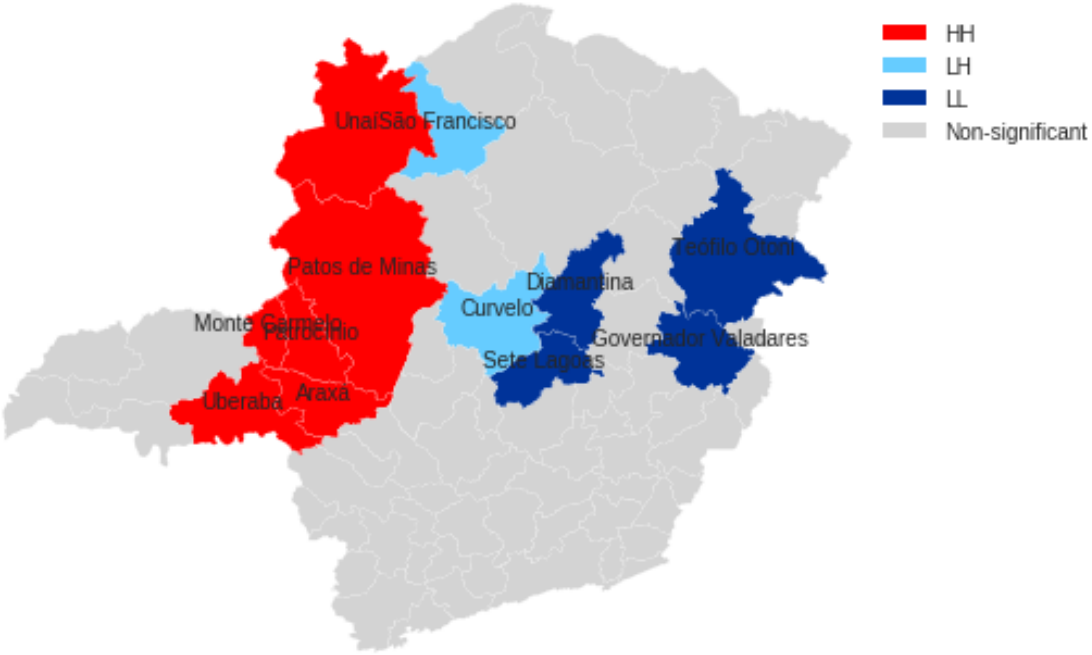
2010


Espinosa
Unaí
Patos de Minas
Araxá
Ipatinga
HH
LH
LL
Non-significant

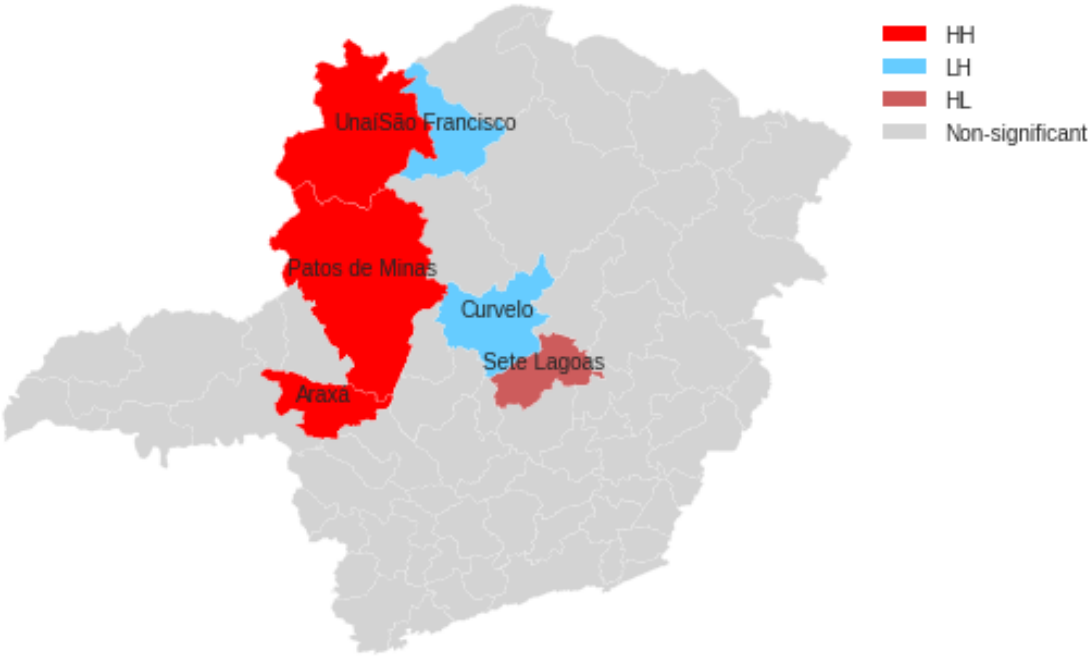
2011


Espinosa
Unaí
Almenara
Teófilo Otoni
Patos de Minas
Curvelo
Governador Valadares
Sete Lagoas
Araxá
HH
LH
LL
Non-significant

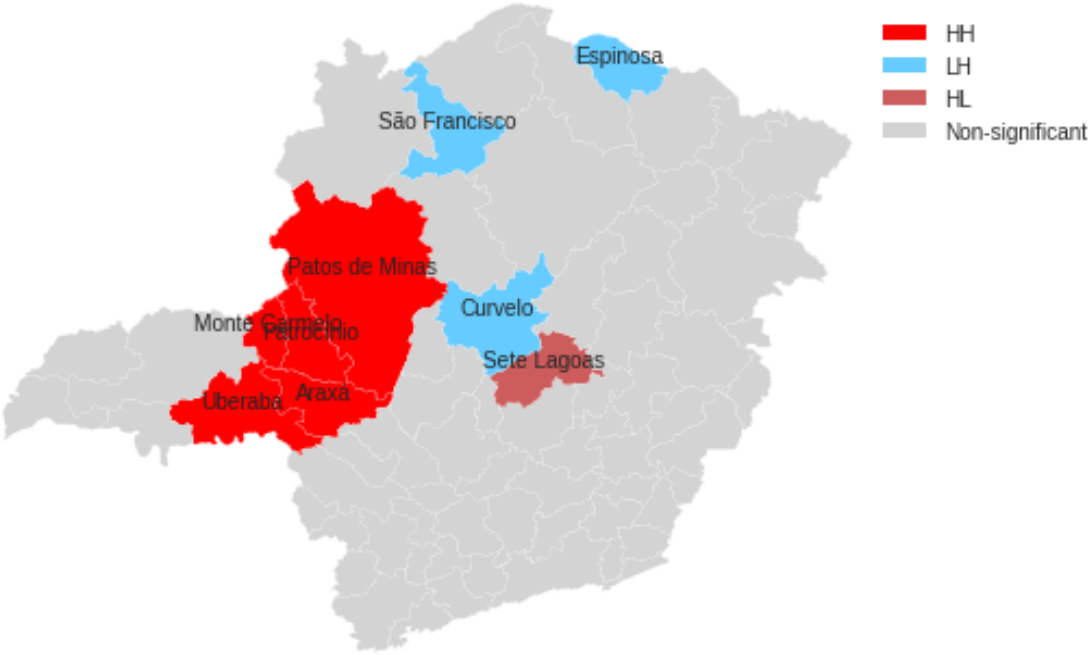
2012


HH
LH
LL
Non-significant
Unaí
São Francisco
Patos de Minas
Monte Carmelo
Patrocínio
Uberaba
Araxá
Diamantina
Curvelo
Sete Lagoas
Teófilo Otoni
Governador Valadares

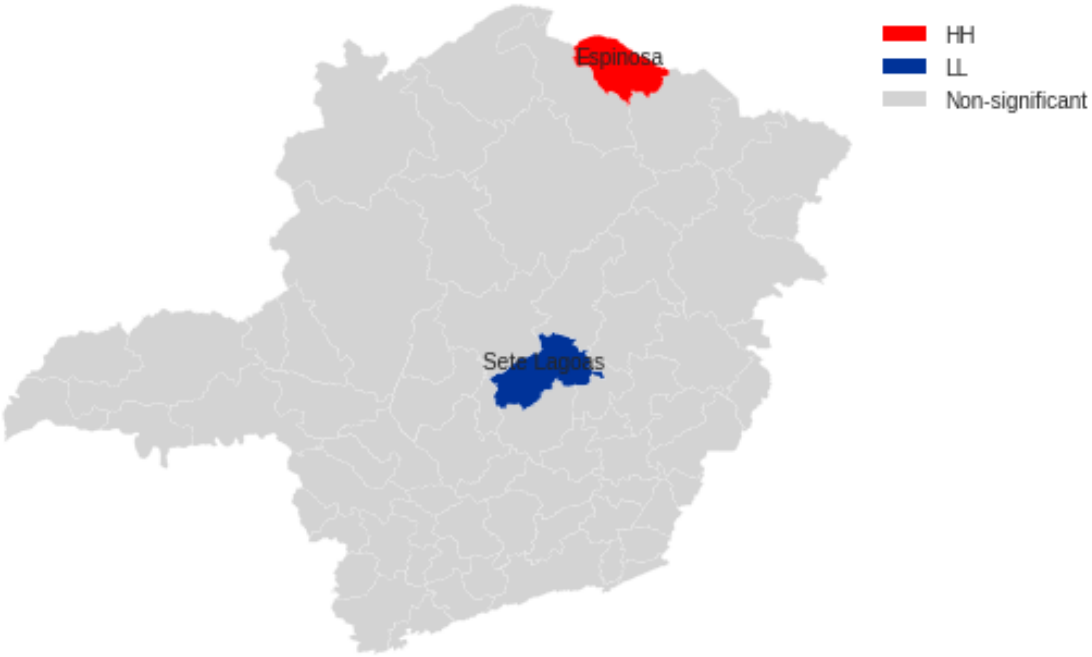
2013


HH
LH
HL
Non-significant
Unaí
São Francisco
Patos de Minas
Araxá
Curvelo
Sete Lagoas

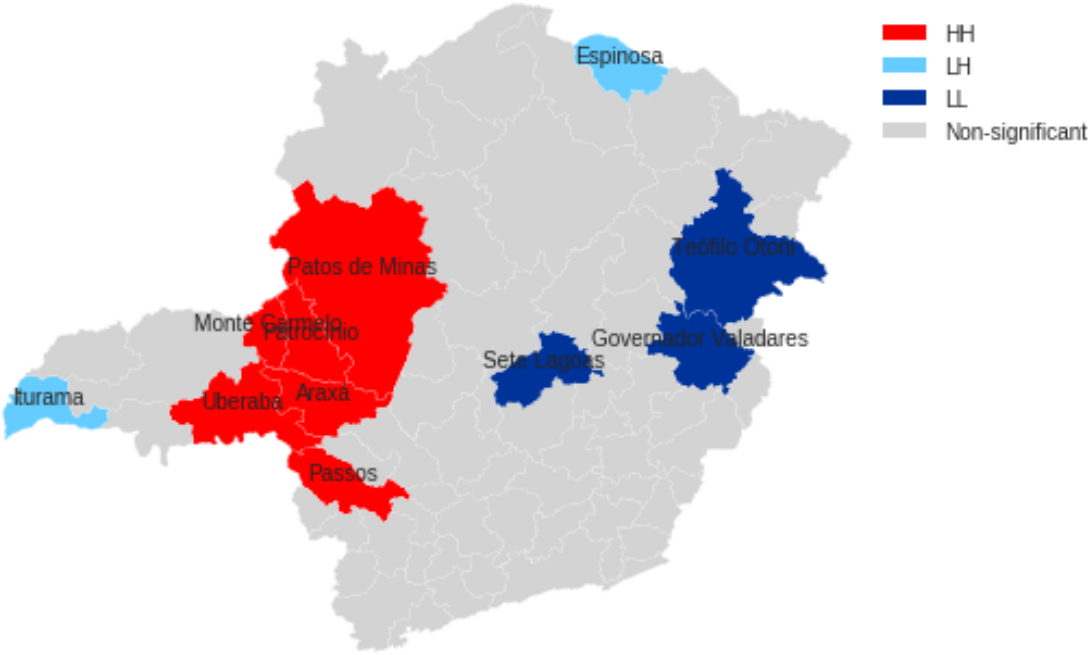
2014


Espinosa
São Francisco
Patos de Minas
Monte Carmelo
Patrocínio
Curvelo
Sete Lagoas
Uberaba
Araxá
HH
LH
HL
Non-significant

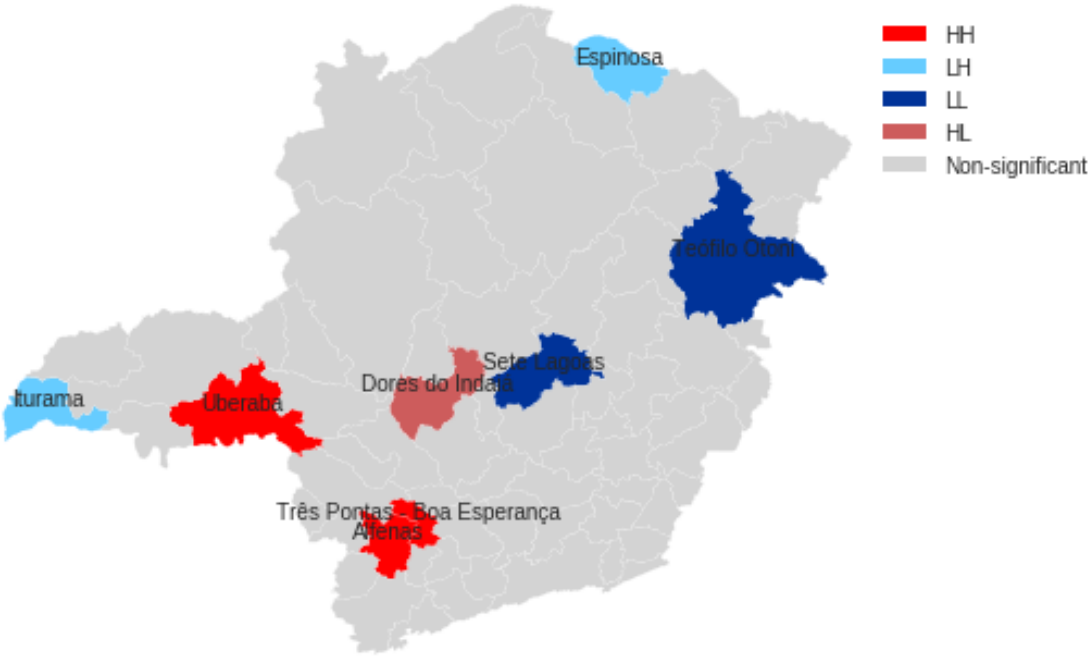
2015


Espinosa
Sete Lagoas
HH
LL
Non-significant

2016


HH
LH
LL
Non-significant
Espinosa
Teófilo Otoni
Patos de Minas
Monte Carmelo
Patrocínio
Governador Valadares
Sete Lagoas
Araxá
Uberaba
Iturama
Passos

2017


HH
LH
LL
HL
Non-significant
Espinosa
Teófilo Otoni
Sete Lagoas
Dores do Indaiá
Iturama
Uberaba
Três Pontas - Boa Esperança
Alfenas