Econometria espacial com R - Aula 02

Unicamp, julho de 2017

Raphael Saldanha, Eduardo Almeida

Análise Exploratória de Dados Espaciais

Leitura do shapefile

Para a leitura de arquivos **shapefile** no R, precisamos usar alguns pacotes. Após a instalação dos pacotes, use os seguintes comandos.

```r
# Pacotes
library(rgdal)
```

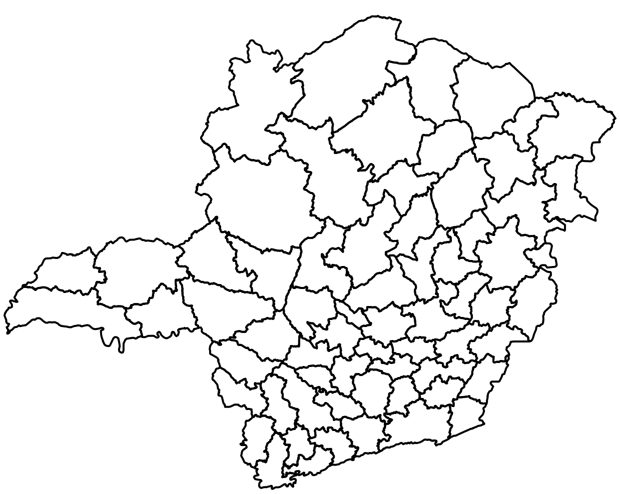
```
## Loading required package: sp
```

```
## rgdal: version: 1.2-8, (SVN revision 663)
##  Geospatial Data Abstraction Library extensions to R successfully loaded
##  Loaded GDAL runtime: GDAL 1.11.3, released 2015/09/16
##  Path to GDAL shared files: /usr/share/gdal/1.11
##  Loaded PROJ.4 runtime: Rel. 4.9.2, 08 September 2015, [PJ_VERSION: 492]
##  Path to PROJ.4 shared files: (autodetected)
##  Linking to sp version: 1.2-5
```

```r
# Abra o arquivo 'gm10.shp'
fp_mg.shp <- readOGR("data", "FP_MG", encoding = "ISO-8859-1")
```

```
## OGR data source with driver: ESRI Shapefile
## Source: "data", layer: "FP_MG"
## with 66 features
## It has 41 fields
```

```r
# Plotar o mapa
plot(fp_mg.shp)
```



Atributos do shapefile

Podemos ver a tabela de atributos do shapefile com desta forma.

```
fp_mg.shp@data
```

```
##    CODMIC                     NOMMIC       Q     AC     AP    K   R  RP RNP
## 0      41                    Aimorés  134310  60132  60132 1833 363 155 208
## 1      49                    Alfenas  564476 113253 113253 7203 312 273  39
## 2      14                   Almenara   84631  20737  20737  783 383 150 233
## 3      55                Andrelândia   59164  27764  27764 1154 264 200  64
## 4      12                    Araçuaí  159897  36779  36779  552 315 162 153
## 5      23                      Araxá  495102 160580 160643 7582 645 564  81
## 6      59                  Barbacena  112632  34487  34508 2208 226 155  71
## 7      30             Belo Horizonte  105895  16925  16959 3048 377 371   5
## 8       9                   Bocaiúva  189720  16802  16802 1261 132 116  16
## 9      26               Bom Despacho 1091182  51871  51871 4195 371 325  46
## 10     45                 Campo Belo   67963  44358  44358 2691 131 131   1
## 11     11                  Capelinha  102870  45530  45543 1317 550 294 256
## 12     40                  Caratinga  126473  78958  78958 2186 214 150  64
## 13     66                 Cataguases   78092  14822  14822 1174 272 272   1
## 14     28 Conceição do Mato Dentro   81280  15569  15569  821 172  17 155
## 15     34      Conselheiro Lafaiete  115865  37330  37539 1193 208 121  87
## 16     25                    Curvelo  303736  41491  41491 2762 534 480  54
## 17     10                 Diamantina   55498  10199  10457  347 326 236  91
## 18     43                Divinópolis  166505  28439  28439 2762 328 305  23
## 19     44                    Formiga  163427  45209  45225 2703 271 246  25
## 20     21                     Frutal 2176070 144431 144870 8472 664 411 253
## 21     37      Governador Valadares  129659  34291  34333 1945 475 313 163
## 22      8                 Grão Mogol   17453  11253  11253  223 134  63  70
## 23     35                   Guanhães   89591  29232  29283 1053 323 247  76
## 24     39                   Ipatinga   26568  14756  14756  562 192 192   1
## 25     31                    Itabira  230597  31504  31764 1729 441 353  88
## 26     32                   Itaguara  104778  19740  19902  989  92  92   1
## 27     56                    Itajubá  110701  24794  24794 1759 156 156   1
## 28     17                  Ituiutaba  230911  69885  69885 5896 307 189 117
## 29      4                    Janaúba   39338  52611  52611 2376 307 259  48
## 30      3                   Januária  238889  47320  47320 2686 738 135 603
## 31     65               Juiz de Fora  128654  39674  40207 2390 376 303  73
## 32     57                     Lavras  100642  40614  40614 3228 111 111   1
## 33     61                   Manhuaçu  172688 126068 126068 3389 310 278  31
## 34     38                    Mantena   29075  31353  31353  410 153  65  88
## 35      7              Montes Claros  184626  65000  65000 4036 520 443  77
## 36     63                     Muriaé  235644  90081  90081 2713 286 286   1
## 37     16                    Nanuque  189983  11379  11379 1111 205 126  78
## 38     46                   Oliveira  122314  46609  46609 2425 237 212  25
## 39     33                 Ouro Preto    9774   5170   5170  252 164 141  23
## 40     29              Pará de Minas   55196   6087   6087 1254 173 173   1
## 41      2                   Paracatu  560903 147391 147391 7781 740 620 120
## 42     47                     Passos 1317059  77890  77939 6582 263 205  59
## 43     20              Patos de Minas  355502  93774  94239 5029 242 206  36
## 44     19                 Patrocínio  447242 141539 141545 8256 577 474 103
## 45     36                    Peçanha   93642  31889  31889  804 249 171  79
## 46     13                 Pedra Azul   30340  11230  11230  304 249 192  57
## 47      6                   Pirapora  179344  33089  33089 2397 443 243 200
```

```
## 48     42                   Pium-í  164970  40279  40587  3734 151 135  16
## 49     51          Poços de Caldas  331145  78842  78842  7148 310 275  35
## 50     60               Ponte Nova 1236183  75393  75408  1663 290 242  48
## 51     52             Pouso Alegre  315758  44605  44605  6007 213 213   1
## 52      5                  Salinas  107322  40394  40394  1248 310 146 164
## 53     53    Santa Rita do Sapucaí  200722  58900  58900  3851 221 221   1
## 54     58         São João Del Rei   72472  29962  29962  2294 302 236  66
## 55     54             São Lourenço   70840  28719  28719  2549 323 323   1
## 56     48 São Sebastião do Paraíso  888840 130314 130314  8712 255 216  39
## 57     27             Sete Lagoas  192550  31532  31532  3575 243 192  51
## 58     15            Teófilo Otôni   84031  46580  46580  1476 448 203 245
## 59     24             Três Marias  433126  26818  26818  1968 366 161 205
## 60     64                      Ubá  443007  48308  48308  1981 274 225  49
## 61     22                  Uberaba 2081419 156150 156150  6500 446 389  56
## 62     18               Uberlândia 1615750 192848 197202 12172 858 822  36
## 63      1                     Unaí  423193 190367 190367  6671 905 342 563
## 64     50                Varginha  514264 165626 165626 10939 386 339  47
## 65     62                   Viçosa  210438  73635  73675  1696 221 120 101
##     DRNP   F    AREA ESCTOT  POPTOT    LP   ACP    KP   ETOTP   DRNPP
## 0     25  84  8354.1    347  152658 0.111 0.394 0.012 0.00227 0.00016
## 1      8   1  4998.9    229  179366 0.111 0.631 0.040 0.00128 0.00004
## 2     15   1 15504.5    384  213342 0.047 0.097 0.004 0.00180 0.00007
## 3     13 227  5047.3    191   70783 0.037 0.392 0.016 0.00270 0.00018
## 4     15   1 10299.4    327  143468 0.143 0.256 0.004 0.00228 0.00010
## 5      6 327 14145.6    182  158275 0.043 1.015 0.048 0.00115 0.00004
## 6     21  90  3370.0    239  204119 0.027 0.169 0.011 0.00117 0.00010
## 7      1 587  5826.9   1531 3622692 0.002 0.005 0.001 0.00042 0.00000
## 8      2 113  7812.3    117   58277 0.055 0.288 0.022 0.00201 0.00003
## 9      6  48  7515.5    165  124687 0.025 0.416 0.034 0.00132 0.00005
## 10     1  98  2714.3    138   95913 0.082 0.462 0.028 0.00144 0.00001
## 11    21   1 12052.2    459  174791 0.090 0.260 0.008 0.00263 0.00012
## 12    11   1  5527.6    314  422952 0.083 0.187 0.005 0.00074 0.00003
## 13     1 125  3932.2    288  198214 0.015 0.075 0.006 0.00145 0.00001
## 14    22   1  6897.7    214   88159 0.033 0.177 0.009 0.00243 0.00025
## 15    30 187  2953.8    231  196023 0.035 0.190 0.006 0.00118 0.00015
## 16     4 268 13792.2    243  136164 0.026 0.305 0.020 0.00178 0.00003
## 17    12   1  7459.2    162   81509 0.064 0.125 0.004 0.00199 0.00015
## 18     5 195  5105.5    267  313674 0.011 0.091 0.009 0.00085 0.00002
## 19     5  96  4577.2    184  134127 0.033 0.337 0.020 0.00137 0.00004
## 20    15   1 16890.8    166  199240 0.026 0.725 0.043 0.00083 0.00008
## 21    14  99 11362.1    476  415877 0.022 0.082 0.005 0.00114 0.00003
## 22     8   1  9108.0    156   39323 0.132 0.286 0.006 0.00397 0.00020
## 23    13   1  5799.5    290  135728 0.046 0.215 0.008 0.00214 0.00010
## 24     1 134  4419.5    291  400130 0.007 0.037 0.001 0.00073 0.00000
## 25    11 230  8022.4    392  328511 0.015 0.096 0.005 0.00119 0.00003
## 26     1  53  2436.2    132   56366 0.152 0.350 0.018 0.00234 0.00002
## 27     1   1  2986.7    226  164325 0.057 0.151 0.011 0.00138 0.00001
## 28    13   1  8748.8    140  130266 0.031 0.536 0.045 0.00107 0.00010
## 29     3 177 15889.4    453  282166 0.076 0.186 0.008 0.00161 0.00001
## 30    18   1 33111.6    443  294247 0.050 0.161 0.009 0.00151 0.00006
## 31     8 150  8946.7    696  673462 0.007 0.059 0.004 0.00103 0.00001
## 32     1 166  3439.8    163  120524 0.058 0.337 0.027 0.00135 0.00001
## 33     6   1  4870.4    392  259913 0.217 0.485 0.013 0.00151 0.00002
## 34    48   1  1857.0    138   82645 0.113 0.379 0.005 0.00167 0.00058
```

```
## 35    4 148 21108.4    649  505735 0.041 0.129 0.008 0.00128 0.00001
## 36    1   1  4766.1    376  333271 0.106 0.270 0.008 0.00113 0.00000
## 37    9   1  8495.8    218  124248 0.023 0.092 0.009 0.00175 0.00007
## 38    6 118  4047.5    190  109462 0.075 0.426 0.022 0.00174 0.00005
## 39    7 299  3157.4    172  136946 0.008 0.038 0.002 0.00126 0.00005
## 40    1   1  1771.0    107   92131 0.021 0.066 0.014 0.00116 0.00001
## 41    5   1 35111.0    317  202934 0.027 0.726 0.038 0.00156 0.00002
## 42    8  56  7127.4    192  185533 0.066 0.420 0.035 0.00103 0.00004
## 43    3   1 10773.3    283  199527 0.037 0.470 0.025 0.00142 0.00002
## 44    9 199 12017.0    190  152654 0.073 0.927 0.054 0.00124 0.00006
## 45   17   1  4616.7    177   88090 0.074 0.362 0.009 0.00201 0.00019
## 46   11   1  5100.0    190   83200 0.051 0.135 0.004 0.00228 0.00013
## 47    9 102 23111.6    266  165475 0.030 0.200 0.014 0.00161 0.00005
## 48    2 113  7666.9    136   73096 0.058 0.551 0.051 0.00186 0.00003
## 49    7  14  4644.2    334  268635 0.090 0.293 0.027 0.00124 0.00003
## 50   10  65  4888.7    316  194911 0.125 0.387 0.009 0.00162 0.00005
## 51    1   1  4931.1    339  239270 0.104 0.186 0.025 0.00142 0.00000
## 52    9   1 17883.1    364  218731 0.121 0.185 0.006 0.00166 0.00004
## 53    1   1  3299.1    206  113804 0.130 0.518 0.034 0.00181 0.00001
## 54   11 119  5787.8    241  153454 0.027 0.195 0.015 0.00157 0.00007
## 55    1 125  3687.4    287  171609 0.054 0.167 0.015 0.00167 0.00001
## 56    8  43  5159.7    288  224264 0.134 0.581 0.039 0.00128 0.00004
## 57    6 134  8560.4    393  286428 0.021 0.110 0.012 0.00137 0.00002
## 58   21   1 11649.1    436  302514 0.066 0.154 0.005 0.00144 0.00007
## 59   19   1 10541.7    132   78789 0.020 0.340 0.025 0.00168 0.00024
## 60   14  79  3603.4    294  211140 0.050 0.229 0.009 0.00139 0.00007
## 61    6 180  9392.6    174  242195 0.015 0.645 0.027 0.00072 0.00002
## 62    2 183 18864.2    337  587376 0.024 0.328 0.021 0.00057 0.00000
## 63   20   1 27653.2    350  126817 0.042 1.501 0.053 0.00276 0.00016
## 64    6  98  7621.5    420  353902 0.104 0.468 0.031 0.00119 0.00002
## 65   21  60  4839.7    321  199267 0.136 0.370 0.009 0.00161 0.00011
##          FP     RNPP      RPP      R_P       QP   X_COORD   Y_COORD CMICRO
## 0  0.000550 0.001363 0.001015 0.002378 0.879810 -41.38876 -19.53116  31041
## 1  0.000006 0.000217 0.001522 0.001739 3.147062 -46.01852 -21.37551  31049
## 2  0.000005 0.001092 0.000703 0.001795 0.396692 -40.65978 -16.35385  31014
## 3  0.003207 0.000904 0.002826 0.003730 0.835850 -44.44562 -21.95045  31055
## 4  0.000007 0.001066 0.001129 0.002196 1.114513 -41.87023 -17.00645  31012
## 5  0.002066 0.000512 0.003563 0.004075 3.128112 -46.95948 -19.59779  31023
## 6  0.000441 0.000348 0.000759 0.001107 0.551796 -43.75469 -21.22165  31059
## 7  0.000162 0.000001 0.000102 0.000104 0.029231 -44.03464 -19.90697  31030
## 8  0.001939 0.000275 0.001990 0.002265 3.255487 -43.78045 -17.38823  31009
## 9  0.000385 0.000369 0.002607 0.002975 8.751369 -45.44562 -19.69013  31026
## 10 0.001022 0.000010 0.001366 0.001366 0.708590 -45.34544 -20.91944  31045
## 11 0.000006 0.001465 0.001682 0.003147 0.588531 -42.66331 -17.41202  31011
## 12 0.000002 0.000151 0.000355 0.000506 0.299024 -42.11393 -19.58605  31040
## 13 0.000631 0.000005 0.001372 0.001372 0.393978 -42.58139 -21.60452  31066
## 14 0.000011 0.001758 0.000193 0.001951 0.921971 -43.37965 -18.79853  31028
## 15 0.000954 0.000444 0.000617 0.001061 0.591079 -43.93158 -20.68153  31034
## 16 0.001968 0.000397 0.003525 0.003922 2.230663 -44.51134 -18.33164  31025
## 17 0.000012 0.001116 0.002895 0.004000 0.680882 -43.59000 -18.05094  31010
## 18 0.000622 0.000073 0.000972 0.001046 0.530822 -44.97872 -20.12036  31043
## 19 0.000716 0.000186 0.001834 0.002020 1.218450 -45.42626 -20.41081  31044
## 20 0.000005 0.001270 0.002063 0.003333       NA -49.80182 -19.80835  31021
## 21 0.000238 0.000392 0.000753 0.001142 0.311772 -41.86191 -18.60464  31037
```

```
## 22 0.000025 0.001780 0.001602 0.003408 0.443837 -42.99130 -16.68005  31008
## 23 0.000007 0.000560 0.001820 0.002380 0.660078 -42.78099 -18.64436  31035
## 24 0.000335 0.000002 0.000480 0.000480 0.066398 -42.60238 -19.36725  31039
## 25 0.000700 0.000268 0.001075 0.001342 0.701946 -43.15195 -19.64321  31031
## 26 0.000940 0.000018 0.001632 0.001632 1.858887 -44.28952 -20.37468  31032
## 27 0.000006 0.000006 0.000949 0.000949 0.673671 -45.49678 -22.41181  31056
## 28 0.000008 0.000898 0.001451 0.002357 1.772611 -49.96062 -18.92139  31017
## 29 0.000627 0.000170 0.000918 0.001088 0.139414 -43.29952 -15.48241  31004
## 30 0.000003 0.002049 0.000459 0.002508 0.811866 -44.70479 -15.31081  31003
## 31 0.000223 0.000108 0.000450 0.000558 0.191034 -43.53889 -21.74364  31065
## 32 0.001377 0.000008 0.000921 0.000921 0.835037 -44.92656 -21.35041  31057
## 33 0.000004 0.000119 0.001070 0.001193 0.664407 -41.99838 -20.21609  31061
## 34 0.000012 0.001065 0.000786 0.001851 0.351806 -41.23444 -18.63929  31038
## 35 0.000293 0.000152 0.000876 0.001028 0.365065 -44.13158 -16.31999  31007
## 36 0.000003 0.000003 0.000858 0.000858 0.707064 -42.27907 -20.90115  31063
## 37 0.000008 0.000628 0.001014 0.001650 1.529063 -40.70051 -17.40168  31016
## 38 0.001078 0.000228 0.001937 0.002165 1.117411 -44.69935 -20.82619  31046
## 39 0.002183 0.000168 0.001030 0.001198 0.071371 -43.51881 -20.38169  31033
## 40 0.000011 0.000011 0.001878 0.001878 0.599103 -44.67568 -19.70902  31029
## 41 0.000005 0.000591 0.003055 0.003647 2.763968 -46.36626 -17.59131  31002
## 42 0.000302 0.000318 0.001105 0.001418 7.098786 -46.63179 -20.63373  31047
## 43 0.000005 0.000180 0.001032 0.001213 1.781724 -46.31133 -18.95346  31020
## 44 0.001304 0.000675 0.003105 0.003780 2.929776 -47.17611 -18.62338  31019
## 45 0.000011 0.000897 0.001941 0.002827 1.063026 -42.44873 -18.26842  31036
## 46 0.000012 0.000685 0.002308 0.002993 0.364663 -41.42159 -16.23048  31013
## 47 0.000616 0.001209 0.001469 0.002677 1.083813 -45.27636 -17.07284  31006
## 48 0.001546 0.000219 0.001847 0.002066 2.256895 -46.28302 -20.10513  31042
## 49 0.000052 0.000130 0.001024 0.001154 1.232695 -46.40565 -22.04879  31051
## 50 0.000333 0.000246 0.001242 0.001488 6.342295 -42.67158 -20.26223  31060
## 51 0.000004 0.000004 0.000890 0.000890 1.319672 -46.09288 -22.39564  31052
## 52 0.000005 0.000750 0.000667 0.001417 0.490657 -42.10846 -15.73740  31005
## 53 0.000009 0.000009 0.001942 0.001942 1.763752 -45.68834 -22.12417  31053
## 54 0.000776 0.000430 0.001538 0.001968 0.472272 -44.33427 -21.22873  31058
## 55 0.000728 0.000006 0.001882 0.001882 0.412799 -45.03241 -22.13770  31054
## 56 0.000192 0.000174 0.000963 0.001137 3.963365 -46.73655 -21.14242  31048
## 57 0.000468 0.000178 0.000670 0.000848 0.672246 -44.18320 -19.16204  31027
## 58 0.000003 0.000810 0.000671 0.001481 0.277776 -41.54675 -17.78239  31015
## 59 0.000013 0.002602 0.002043 0.004645 5.497290 -45.23381 -18.70230  31024
## 60 0.000374 0.000232 0.001066 0.001298 2.098167 -43.02563 -21.13436  31064
## 61 0.000743 0.000231 0.001606 0.001841 8.593980 -48.17826 -19.61311  31022
## 62 0.000312 0.000061 0.001399 0.001461 2.750793 -48.55164 -18.95539  31018
## 63 0.000008 0.004439 0.002697 0.007136 3.337037 -46.50819 -15.82971  31001
## 64 0.000277 0.000133 0.000958 0.001091 1.453125 -45.52145 -21.29040  31050
## 65 0.000301 0.000507 0.000602 0.001109 1.056060 -43.00738 -20.81637  31062
##          VP      VPP    LA    LM     L CO_RUR  NU_RUR    CO_TOT      EER
## 0    78349 0.513232 42737  3997 46734   4937 1314045   4402218 0.003757
## 1   216064 1.204598 33737  1118 34855   6736 2957879  12228817 0.002277
## 2    40164 0.188261 30920  4259 35179   1431  282421   3363297 0.005067
## 3    42360 0.598449 15135   908 16043   1717  498819   2819950 0.003442
## 4    28864 0.201188 45015  7069 52084   1431  365829   2404595 0.003912
## 5   197010 1.244732 23442  1488 24930   4357 1476959  14220985 0.002950
## 6    54692 0.267942 19557   895 20452   3360 1394757  14265916 0.002409
## 7    89076 0.024588 25300  1927 27227   5609 3921436 413082467 0.001430
## 8    20282 0.348028 12130  1897 14027   1369  299957   2588858 0.004564
```

```
## 9   128541 1.030909 19085   940 20025   4584 1960540   9452629 0.002338
## 10   55820 0.581986 15829   612 16441   2793  686604   5756217 0.004068
## 11   65463 0.374522 50416  6980 57396    936  404033   2900130 0.002317
## 12   89075 0.210603 58010  8585 66595   5423 1725161   6953826 0.003143
## 13   53063 0.267706 18383  1551 19934    264  141303    455052 0.001868
## 14   27311 0.309793 14709  1467 16176    967  195666   1590011 0.004942
## 15   27079 0.138142 19942  2014 21956   2847  665610  13274584 0.004277
## 16   70487 0.517663 19058  1449 20507   4086 1541372   8151863 0.002651
## 17   15489 0.190028 11021  2090 13111    637  112293   3892289 0.005673
## 18  138719 0.442239 21279  1649 22928   7046 2403431  39886820 0.002932
## 19   73013 0.544357 18725  1183 19908   4502 1189319  13790709 0.003785
## 20  210274 1.055380 30998  2156 33154   6327 1941451  11075190 0.003259
## 21   88727 0.213349 38512  4917 43429   4922 2044539  27873843 0.002407
## 22   30990 0.788088 15756  4304 20060    628   85152    423941 0.007375
## 23   36903 0.271889 26525  3069 29594   1648  439844   3289883 0.003747
## 24   28489 0.071199 12907   843 13750   1519  403964  27282160 0.003760
## 25   64518 0.196395 23610  1660 25270   4083 1150034  19352349 0.003550
## 26   29966 0.531633 20304  2394 22698   3702  690868   3168928 0.005358
## 27   50277 0.305961 22145   816 22961   3353  969573  10712985 0.003458
## 28  110121 0.845355 16365  1048 17413   3961 1389225  11807580 0.002851
## 29   47580 0.168624 55451  7368 62819   8486 3293492   7193553 0.002577
## 30   64563 0.219418 68306 13654 81960   1574  784375   5097300 0.002007
## 31   90106 0.133795 31638  1502 33140   4592 2154379  55508585 0.002131
## 32   76466 0.634446 15789  1044 16833   3036 1304031   8732170 0.002328
## 33  149204 0.574054 68528  9721 78249   1861  555615   1072798 0.003349
## 34   30645 0.370803 16101  1823 17924   1308  421891   1689751 0.003100
## 35  103445 0.204544 76822 11579 88401   7584 2470366  27277342 0.003070
## 36  110726 0.332240 53601  5721 59322   3878 1199394   3369723 0.003233
## 37   56910 0.458036 18212  2406 20618   1788  518968   4785273 0.003445
## 38   81776 0.747072 23944  1632 25576   3680 1197559   5064042 0.003073
## 39    9485 0.069261  4865   388  5253   1737  497061  11231499 0.003495
## 40   99125 1.075914  9545   409  9954   2334 2011751  13450028 0.001160
## 41  230551 1.136089 34094  3347 37441   5388 2914647  15077819 0.001849
## 42  169796 0.915180 31824  1578 33402   5983 2310669  16712601 0.002589
## 43  216031 1.082716 35835  3513 39348   6914 2353333  14606742 0.002938
## 44  313068 2.050834 31485  2127 33612   4833 2379711  13625031 0.002031
## 45   37237 0.422715 23135  3589 26724   1010  268844   1259711 0.003757
## 46   13296 0.159808 14379  1979 16358    884  187243   3213671 0.004721
## 47   53726 0.324677 20893  2603 23496   1333  673624   9908574 0.001979
## 48   77251 1.056843 19978  2016 21994   2854 1094491   3847919 0.002608
## 49  161867 0.602554 43611  3813 47424   7829 2847215  10179976 0.002750
## 50  112400 0.576673 45530  3589 49119   4565 2217617   7574039 0.002059
## 51  125620 0.525014 46813  1804 48617   3588 1027005  17951689 0.003494
## 52   42462 0.194129 57097  8846 65943   1200  337341   3063357 0.003557
## 53  107059 0.940731 28909   928 29837   3744 1692044   7016582 0.002213
## 54   57933 0.377527 21932  1781 23713   3734  721567   8689179 0.005175
## 55   95474 0.556346 24375  1683 26058   3743 1483975  12569932 0.002522
## 56  197295 0.879744 49587  2992 52579   6247 2885781  14325629 0.002165
## 57  111072 0.387783 25146  1725 26871   5090 2844565  45447487 0.001789
## 58   58989 0.194996 45492  6165 51657   1726  407798   1731775 0.004232
## 59   81372 1.032784 12820   533 13353   2311  913193   4814997 0.002531
## 60   66805 0.316401 33628  3465 37093    337   70051    337056 0.004811
## 61  155791 0.643246 14704  1613 16317   4073 2493861  33715991 0.001633
## 62  327458 0.557493 39445  3135 42580   7861 4338184  71831290 0.001812
```

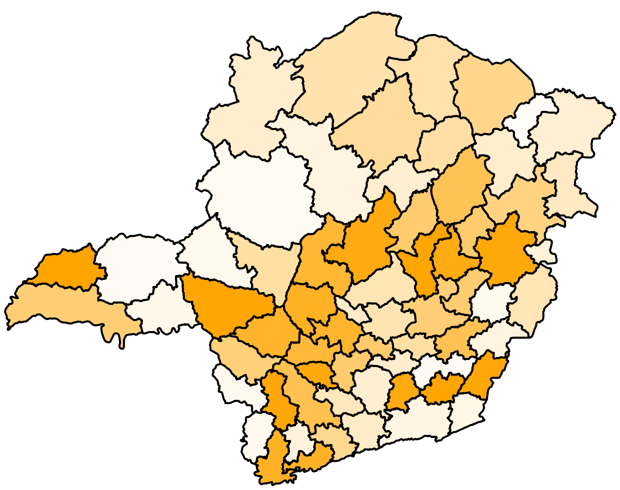
```
## 63 146648 1.156375 30007  3930 33937   2286  844006   5709997 0.002709
## 64 293247 0.828611 53903  2731 56634   8351 4521924  35314099 0.001847
## 65  75418 0.378477 51070  5539 56609   2788  703010   6234143 0.003966
##          K_L VEG TEMP PREC     KL
## 0  0.039222   1    4    1 0.0392
## 1  0.206656   2    2    3 0.2067
## 2  0.022258   2    5    1 0.0223
## 3  0.071932   2    1    4 0.0719
## 4  0.010598   3    4    1 0.0106
## 5  0.304132   1    2    4 0.3041
## 6  0.107960   2    1    3 0.1080
## 7  0.111948   1    2    3 0.1119
## 8  0.089898   1    4    2 0.0899
## 9  0.209488   1    3    3 0.2095
## 10 0.163676   1    2    3 0.1637
## 11 0.022946   2    4    3 0.0229
## 12 0.032825   2    4    3 0.0328
## 13 0.058894   2    4    3 0.0589
## 14 0.050754   2    2    4 0.0508
## 15 0.054336   2    1    3 0.0543
## 16 0.134686   1    3    2 0.1347
## 17 0.026466   1    1    2 0.0265
## 18 0.120464   1    2    3 0.1205
## 19 0.135775   1    2    3 0.1358
## 20 0.255535   1    4    4 0.2555
## 21 0.044786   2    4    1 0.0448
## 22 0.011117   3    3    3 0.0111
## 23 0.035582   2    2    3 0.0356
## 24 0.040873   2    4    3 0.0409
## 25 0.068421   2    2    3 0.0684
## 26 0.043572   1    2    3 0.0436
## 27 0.076608   2    2    3 0.0766
## 28 0.338598   1    5    4 0.3386
## 29 0.037823   3    4    1 0.0378
## 30 0.032772   1    4    1 0.0328
## 31 0.072118   2    2    4 0.0721
## 32 0.191766   1    2    3 0.1918
## 33 0.043310   2    2    3 0.0433
## 34 0.022874   2    5    1 0.0229
## 35 0.045656   1    4    2 0.0457
## 36 0.045733   2    2    3 0.0457
## 37 0.053885   2    5    3 0.0539
## 38 0.094815   1    2    3 0.0948
## 39 0.047973   2    1    3 0.0480
## 40 0.125980   1    2    3 0.1260
## 41 0.207820   1    4    3 0.2078
## 42 0.197054   1    3    3 0.1971
## 43 0.127808   1    2    3 0.1278
## 44 0.245627   1    3    4 0.2456
## 45 0.030085   2    3    3 0.0301
## 46 0.018584   2    4    3 0.0186
## 47 0.102017   1    4    2 0.1020
## 48 0.169774   1    3    3 0.1698
## 49 0.150725   2    1    1 0.1507
```

```
## 50 0.033857   2    2    3 0.0339
## 51 0.123558   2    1    2 0.1236
## 52 0.018925   3    3    1 0.0189
## 53 0.129068   2    1    3 0.1291
## 54 0.096740   2    1    3 0.0967
## 55 0.097820   2    1    3 0.0978
## 56 0.165694   2    1    3 0.1657
## 57 0.133043   1    2    3 0.1330
## 58 0.028573   2    4    2 0.0286
## 59 0.147383   1    4    3 0.1474
## 60 0.053406   2    3    3 0.0534
## 61 0.398358   1    3    3 0.3984
## 62 0.285862   1    3    3 0.2859
## 63 0.196570   1    4    3 0.1966
## 64 0.193153   1    2    3 0.1932
## 65 0.029960   2    2    3 0.0300
```

Mapa

Podemos produzir um mapa colorido com os seguintes comandos.

```
p <- colorRampPalette(c("white", "orange"))(128)
palette(p)
plot(fp_mg.shp, col = fp_mg.shp@data$Q)
```



Sua vez

Faça um mapa com a variável AC com a cor vermelha.

Matriz de vizinhos espaciais

Para a criação de matrizes de vizinhos espaciais, iremos utilizar o pacote `spdep`.

```
# Pacote
library(spdep)
```

```
## Loading required package: Matrix
```

Matriz queen e rook

```r
# Matriz queen
w1 <- nb2listw(poly2nb(fp_mg.shp, queen = TRUE))
summary(w1)
```

```
## Characteristics of weights list object:
## Neighbour list object:
## Number of regions: 66
## Number of nonzero links: 336
## Percentage nonzero weights: 7.713499
## Average number of links: 5.090909
## Link number distribution:
##
##  2  3  4  5  6  7  8  9
##  2  9 12 16 16  8  2  1
## 2 least connected regions:
## 28 37 with 2 links
## 1 most connected region:
## 64 with 9 links
##
## Weights style: W
## Weights constants summary:
##    n   nn S0       S1       S2
## W 66 4356 66 27.58858 269.8006
```

```r
# Matrix queen padronizada na linha
w1.w <- nb2listw(poly2nb(fp_mg.shp, queen=TRUE), style="W")
summary(w1.w)
```

```
## Characteristics of weights list object:
## Neighbour list object:
## Number of regions: 66
## Number of nonzero links: 336
## Percentage nonzero weights: 7.713499
## Average number of links: 5.090909
## Link number distribution:
##
##  2  3  4  5  6  7  8  9
##  2  9 12 16 16  8  2  1
## 2 least connected regions:
## 28 37 with 2 links
## 1 most connected region:
## 64 with 9 links
##
## Weights style: W
## Weights constants summary:
##    n   nn S0       S1       S2
## W 66 4356 66 27.58858 269.8006
```

```r
# Matriz rook
w2 <- nb2listw(poly2nb(fp_mg.shp, queen = FALSE))
summary(w2)
```

```
## Characteristics of weights list object:
```

```
## Neighbour list object:
## Number of regions: 66
## Number of nonzero links: 332
## Percentage nonzero weights: 7.621671
## Average number of links: 5.030303
## Link number distribution:
##
##  2  3  4  5  6  7  8
##  2  9 12 18 15  7  3
## 2 least connected regions:
## 28 37 with 2 links
## 3 most connected regions:
## 11 25 64 with 8 links
##
## Weights style: W
## Weights constants summary:
##    n   nn S0       S1       S2
## W 66 4356 66 27.82221 269.6778
```

```
# Matriz rook padronizada globalmente
w2.c <- nb2listw(poly2nb(fp_mg.shp, queen = FALSE), style = "C")
summary(w2.c)
```

```
## Characteristics of weights list object:
## Neighbour list object:
## Number of regions: 66
## Number of nonzero links: 332
## Percentage nonzero weights: 7.621671
## Average number of links: 5.030303
## Link number distribution:
##
##  2  3  4  5  6  7  8
##  2  9 12 18 15  7  3
## 2 least connected regions:
## 28 37 with 2 links
## 3 most connected regions:
## 11 25 64 with 8 links
##
## Weights style: C
## Weights constants summary:
##    n   nn S0       S1      S2
## C 66 4356 66 26.24096 285.489
```

Distância inversa

```
coords <- coordinates(fp_mg.shp)
nb <- dnearneigh(coords, 0, 1000)
dlist <- nbdists(nb, coords)
dlist <- lapply(dlist, function(x) 1/x)
w3 <- nb2listw(nb, glist=dlist)
summary(w3)
```

```
## Characteristics of weights list object:
## Neighbour list object:
```

```
## Number of regions: 66
## Number of nonzero links: 4290
## Percentage nonzero weights: 98.48485
## Average number of links: 65
## Link number distribution:
##
## 65
## 66
## 66 least connected regions:
## 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 31 32 33 34 35 36 3
## 66 most connected regions:
## 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 31 32 33 34 35 36 3
##
## Weights style: W
## Weights constants summary:
##    n   nn S0       S1       S2
## W 66 4356 66 3.016494 266.1162
```

```r
# Distância inversa padronizada pelo número de vizinhos
w3.u <- nb2listw(nb, glist=dlist, style="U")
summary(w3.u)
```

```
## Characteristics of weights list object:
## Neighbour list object:
## Number of regions: 66
## Number of nonzero links: 4290
## Percentage nonzero weights: 98.48485
## Average number of links: 65
## Link number distribution:
##
## 65
## 66
## 66 least connected regions:
## 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 31 32 33 34 35 36 3
## 66 most connected regions:
## 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 31 32 33 34 35 36 3
##
## Weights style: U
## Weights constants summary:
##    n   nn S0           S1         S2
## U 66 4356  1 0.0007459454 0.06384681
```

Para ver mais opções, veja a ajuda deste comando: `?nb2listw`

Matriz de k-vizinhos espaciais

A escolha do número ideal de k vizinhos será realizada testando-se vários k e utilizando-se o que retornou o maior valor para a estatística I de Moran significativo.

```r
# Número de permutações
per <- 999

# Número máximo de k vizinhos testados
kv <- 20
```

```r
# Nome dos registros
IDs <- row.names(fp_mg.shp@data)

# Criação da tabela que irá receber a estatística I de Moran e significância para cada k testado
res.pesos <- data.frame(k=numeric(),i=numeric(),valorp=numeric())

# Início do loop
for(k in 1:kv)
{
  # Armazenando número k atual
  res.pesos[k,1] <- k
  # Calculando o I e significância para o k atual
  moran.k <- moran.mc(fp_mg.shp@data$Q,
                      listw=nb2listw(knn2nb(
                      knearneigh(coords, k=k),
                      row.names=IDs),style="B"),
                      nsim=per)
  # Armazenando o valor I para o k atual
  res.pesos[k,2] <- moran.k$statistic
  # Armazenando o p-value para o k atual
  res.pesos[k,3] <- moran.k$p.value
}

# Ver a tabela de k vizinhos, I de Moran e significância
res.pesos
```

```
##     k         i valorp
## 1   1 0.5228074  0.002
## 2   2 0.3875458  0.003
## 3   3 0.4531317  0.001
## 4   4 0.4199339  0.001
## 5   5 0.3944831  0.001
## 6   6 0.3595862  0.001
## 7   7 0.3461349  0.001
## 8   8 0.3286129  0.001
## 9   9 0.3064023  0.001
## 10 10 0.3157462  0.001
## 11 11 0.3028398  0.001
## 12 12 0.2942354  0.001
## 13 13 0.2791438  0.001
## 14 14 0.2620697  0.001
## 15 15 0.2541920  0.001
## 16 16 0.2429784  0.001
## 17 17 0.2320723  0.001
## 18 18 0.2213339  0.001
## 19 19 0.2117356  0.001
## 20 20 0.2017898  0.001
```

```r
# Sendo todos significativos, iremos usar o k que retornou o maior valor I
maxi <- which.max(res.pesos[,2])

# Criação da matriz usando o k escolhido
w5 <- nb2listw(knn2nb(knearneigh(coords, k=maxi),row.names=IDs),style="B")
summary(w5)
```

```
## Characteristics of weights list object:
## Neighbour list object:
## Number of regions: 66
## Number of nonzero links: 66
## Percentage nonzero weights: 1.515152
## Average number of links: 1
## Non-symmetric neighbours list
## Link number distribution:
##
##  1
## 66
## 66 least connected regions:
## 0 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 31 32 33 34 35 36
## 66 most connected regions:
## 0 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 31 32 33 34 35 36
##
## Weights style: B
## Weights constants summary:
##    n   nn S0 S1  S2
## B 66 4356 66 98 308
```

Autocorrelação espacial global

I de Moran

```r
moran.test(fp_mg.shp@data$Q, listw = w5)
```

```
##
##  Moran I test under randomisation
##
## data:  fp_mg.shp@data$Q
## weights: w5
##
## Moran I statistic standard deviate = 3.8645, p-value = 5.566e-05
## alternative hypothesis: greater
## sample estimates:
## Moran I statistic       Expectation          Variance
##        0.52280745       -0.01538462        0.01939499
```

```r
moran.mc(fp_mg.shp@data$Q, listw = w5, nsim = 999)
```

```
##
##  Monte-Carlo simulation of Moran I
##
## data:  fp_mg.shp@data$Q
## weights: w5
## number of simulations + 1: 1000
##
## statistic = 0.52281, observed rank = 998, p-value = 0.002
## alternative hypothesis: greater
```

C de Geary

```r
geary.test(fp_mg.shp@data$Q, listw = w5)
```

```
##
##  Geary C test under randomisation
##
## data:  fp_mg.shp@data$Q
## weights: w5
##
## Geary C statistic standard deviate = 2.6176, p-value = 0.004428
## alternative hypothesis: Expectation greater than statistic
## sample estimates:
## Geary C statistic       Expectation          Variance
##        0.46130049        1.00000000        0.04235442
```

G de Getis-Ord

```r
globalG.test(fp_mg.shp@data$Q, listw = w5)
```

```
##
##  Getis-Ord global G statistic
##
## data:  fp_mg.shp@data$Q
## weights: w5
##
## standard deviate = 3.2113, p-value = 0.0006607
## alternative hypothesis: greater
## sample estimates:
## Global G statistic        Expectation           Variance
##       3.071155e-02       1.538462e-02       2.277991e-05
```

Autocorrelação espacial local

G de Gettis-Ords

```r
lg1 <- localG(fp_mg.shp@data$Q, listw = w5)
summary(lg1)
```

```
##     Min.  1st Qu.   Median     Mean  3rd Qu.     Max.
## -0.69900 -0.53552 -0.44190 -0.04454 -0.06779  4.21616
```

I de Moran

```r
# Cálculo
lm1 <- localmoran(fp_mg.shp@data$Q, listw = w5)
summary(lm1)
```

```
##        Ii                 E.Ii               Var.Ii            Z.Ii
##  Min.   :-0.83956   Min.   :-0.01538   Min.   :0.8627   Min.   :-0.88734
##  1st Qu.: 0.07066   1st Qu.:-0.01538   1st Qu.:0.8627   1st Qu.: 0.09264
```

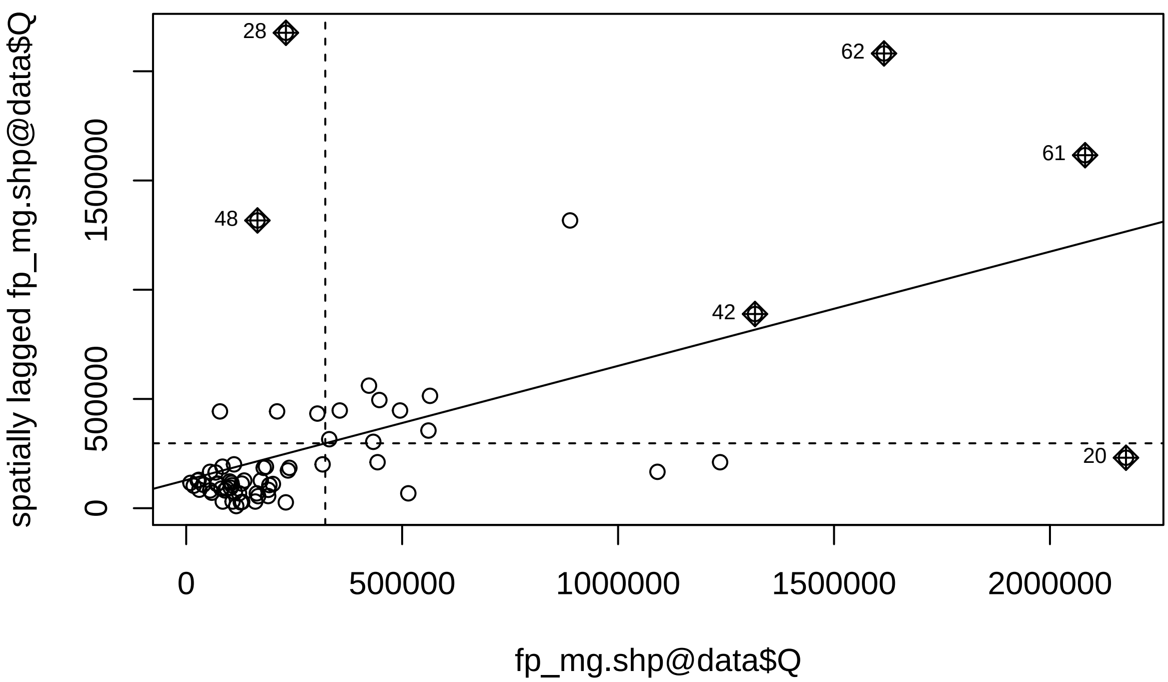
```
##  Median : 0.20075   Median :-0.01538   Median :0.8627   Median : 0.23270
##  Mean   : 0.52281   Mean   :-0.01538   Mean   :0.8627   Mean   : 0.57944
##  3rd Qu.: 0.27959   3rd Qu.:-0.01538   3rd Qu.:0.8627   3rd Qu.: 0.31759
##  Max.   :11.31579   Max.   :-0.01538   Max.   :0.8627   Max.   :12.19963
##    Pr(z > 0)
##  Min.   :0.0000
##  1st Qu.:0.3754
##  Median :0.4080
##  Mean   :0.4216
##  3rd Qu.:0.4631
##  Max.   :0.8126
```

```r
# Quantos são significativos?
table(lm1<0.05)
```

```
##
## FALSE  TRUE
##   231    99
```

Diagrama de dispersão de Moran

```r
moran.plot(fp_mg.shp@data$Q, listw = w5)
```



Sua vez

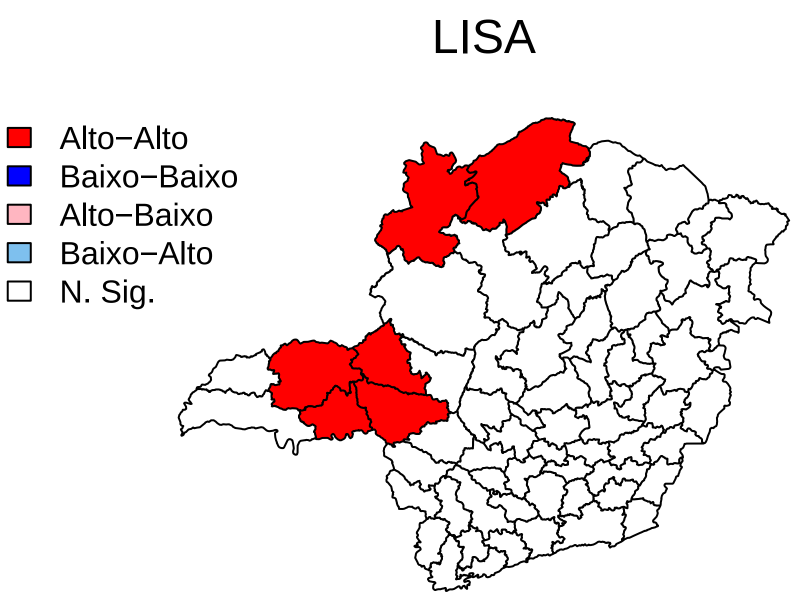
Calcule o I de Moran local usando a matriz de vizinhança `w1` para a variável `AC`e verifique quantas regiões são significativas. Depois, faça o diagrama de dispersão.

LISA map

O R não tem uma função pronta para criar um mapa LISA, então nós criamos abaixo nossa própria função: `lisaplot`. Depois de declarada, uma função pode ser usada repetidamente variando seus argumentos.

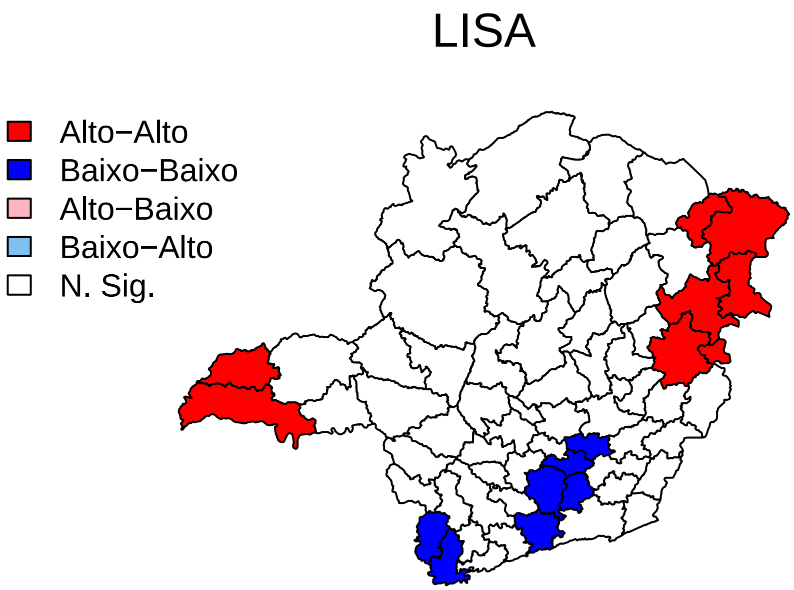
Rode o código abaixo.

```
lisaplot <- function(shapefile, values, listw, pval = 0.05){
  require(spdep)

  svalues <- scale(values)
  lag_svalues <- spdep::lag.listw(listw, svalues)
  locm <- spdep::localmoran(values, listw)
  sig <- rep(5, length(values))

  sig[(svalues >= 0 & lag_svalues >= 0) & (locm[,5] <= pval)] <- 1
  sig[(svalues <= 0 & lag_svalues <= 0) & (locm[,5] <= pval)] <- 2
  sig[(svalues >= 0 & lag_svalues <= 0) & (locm[,5] <= pval)] <- 3
  sig[(svalues >= 0 & lag_svalues <= 0) & (locm[,5] <= pval)] <- 4
  sig[(svalues <= 0 & lag_svalues >= 0) & (locm[,5] <= pval)] <- 5

  breaks <- seq(1, 5, 1)
  labels <- c("Alto-Alto", "Baixo-Baixo", "Alto-Baixo", "Baixo-Alto", "N. Sig.")
  np <- findInterval(sig, breaks)
  colors <- c("red", "blue", "lightpink", "skyblue2", "white")
  plot(shapefile, col = colors[np])  #colors[np] manually sets the color for each county
  mtext("LISA", cex = 1.5, side = 3, line = 1)
  legend("topleft", legend = labels, fill = colors, bty = "n")
}

lisaplot(fp_mg.shp, fp_mg.shp@data$R, w5)
```



E o LISA para a variável `TEMP`.

```
lisaplot(fp_mg.shp, fp_mg.shp@data$TEMP, w5)
```



Sua vez

Faça o LISA para a variável `AP` com a matriz `w1`.