

問題解決技法入門

2. GIS, Voronoi diagram

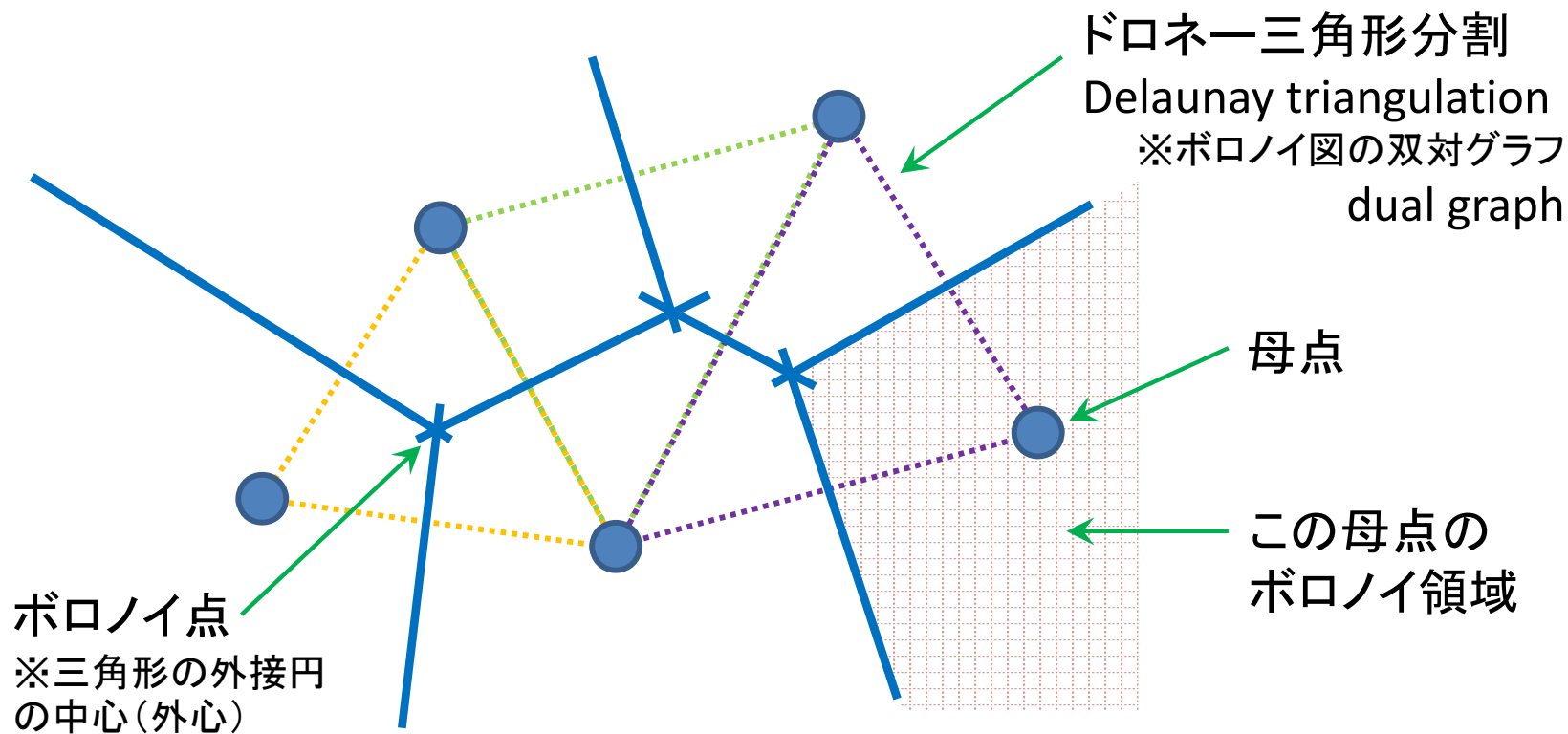
堀田 敬介

※GIS = Geographic Information System(s), 地理情報システム

Voronoi diagram とは？

- ボロノイ図 Voronoi diagram

- 空間上の複数個の点(母点)をもとに領域分割
- 各領域(ボロノイ領域)から最も近い母点はその領域内の母点
- 2次元ユークリッド空間上でユークリッド距離を考えると, 各領域は2点の垂直二等分線で分割される その際, 3つの母点を分ける境界線の交点は1点で交わる(ボロノイ点)



RでVoronoi図を簡単に描く1

- Rを起動し, R Console で以下を入力

```
> install.packages("deldir")
```

← package(deldir) のインストール

```
> library(deldir)
```

← package(deldir) の読み込み

- 母点を適当につくり voronoi図を描く

```
> x <- c(2, 3, 7, 1, 3, 8)
```

← 6点の(x,y)座標設定

```
> y <- c(2, 3, 2, 5, 8, 9)
```

(2,2), (3,3), (7,2), (1,5),...

```
> dd <- deldir(x, y, plot=T)
```

※ <- : 右のモノを左に代入

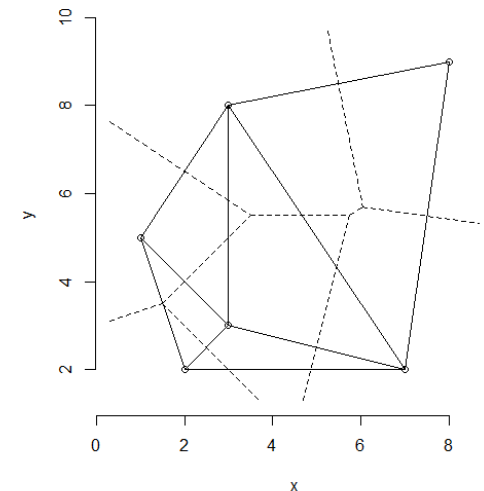
※ c(...) : ベクトルを作成

ボロノイ図 Voronoi diagram【点線】 と
ドロネー三角形分割 Delaunay triangulation【実線】
を描く

```
> tl <- tile.list(dd)
```

```
> plot(tl)
```

ボロノイ図 Voronoi diagram【実線】のみを描く



RでVoronoi図を簡単に描く 2

- 座標を用意

```
> x <- runif(20)
> y <- runif(20)
```

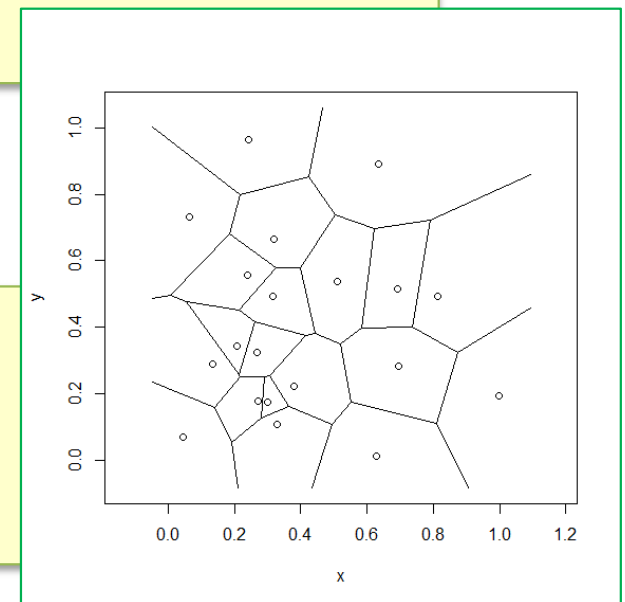
← [0,1)一様乱数を20個生成し,
(x,y)座標とする
※ **runif** : the **un**iform distribution
r = **ran**dom

- 座標の確認

```
> x
[1] 0.04693021 0.20785805 0.32963935 0.51061063 0.81478018 0.13452010
[7] 0.26855775 0.30166600 0.24344253 0.62985486 0.37873759 0.31884216
[13] 0.63481753 0.31565712 0.99787164 0.06323913 0.24069670 0.69282271
[19] 0.27066383 0.69622948
```

- ボロノイ図を描く

```
> dd <- deldir(x, y)
> tl <- tile.list(dd)
> plot(tl)
```



問題解決技法入門

2. GIS, Voronoi diagram

RでVoronoi図を描く【csv編】

RでVoronoi図を描く【csv編0】

- データとしてcsv ファイル (*.csv) を準備
 - 「マイドキュメント(Y:)」に「R」フォルダをつくり保存

“kana.csv” ファイルの中身 (Excelで開いた場合)

name	type	lon	lat
鶴見区	ward	139.6825	35.5084
神奈川区	ward	139.6294	35.4769
西区	ward	139.6170	35.4536
中区	ward	139.6422	35.4447
...	...	...	...

※name: 名前

※type: 種類 (ward = 区, city = 市, ...)

※lon: longitude (経度) の略

※lat: latitude (緯度) の略

ドキュメント
(Y:)

R

kana.csv

※ここに保存

“kana.csv” をテキストエディタ (TeraPadやメモ帳) で開くと...

```
name,type,lon,lat
鶴見区,ward,139.6825,35.5084
神奈川区,ward,139.6294,35.4769
西区,ward,139.617,35.4536
中区,ward,139.6422,35.4447
南区,ward,139.6088,35.4315
保土ヶ谷区,ward,139.596,35.46
磯子区,ward,139.6184,35.4024
金沢区,ward,139.6244,35.3373
港北区,ward,139.633,35.5191
```

※csvファイルとは？

- データがコンマ(,)で区切られたテキストファイルのこと
- csvファイルを開くとExcelが自動で起動し、コンマ(,)で区切られたデータが各セルに表示される
- TeraPad等のテキストエディタで開くと、左のように表示される

RでVoronoi図を描く【csv編1】

• csv ファイル (*.csv) を読み込みVoronoi図を描く

✓ csvファイルの読み込み

※csvファイルの1行目はheader(項目名)が記述されているのでT(True)を指定する

```
> df <- read.csv("Y:/R/kana.csv", header=T)
```

("... ")に指定したcsvファイルを読みという命令
※ファイルはフルパス("Y:/R/kana.csv")で指定
例では、YドライブのRフォルダ内にあるkana.csvという名前のファイルをフルパスで指定している

ファイルを読み込んだ後、変数dfに代入する

✓ 読み込んだファイルの中身確認

```
> df
```

← 変数dfの中身確認は変数名を書いてEnter

✓ Voronoi図を描く

※plot=T(True) とすると描く

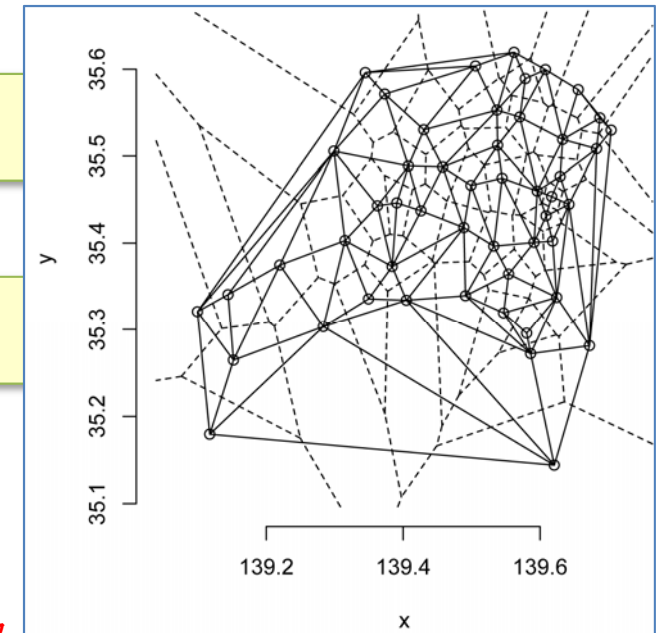
```
> dd <- deldir(df$lon, df$lat, plot=T)
```

指定した(x,y)座標でVoronoi図をつくり、
結果をddに代入

x座標として変数dfのlonを指定
y座標として変数dfのlatを指定

「変数名\$項目名」
と書くと列のデータを意味する
例) df\$lon ... 変数dfの項目lonのデータ

描画したボロノイ図【点線】
とドローネー三角形分割【実線】



RでVoronoi図を描く【csv編2】

- Voronoi図だけを描きたい場合

(ドロネー三角形分割は表示したくない場合)

```
> df <- read.csv("Y:/R/kana.csv", header=T)
```

```
> df
```

変数dfの中身を確認

```
> dd <- deldir(df$lon, df$lat)
```

```
> tl <- tile.list(dd)
```

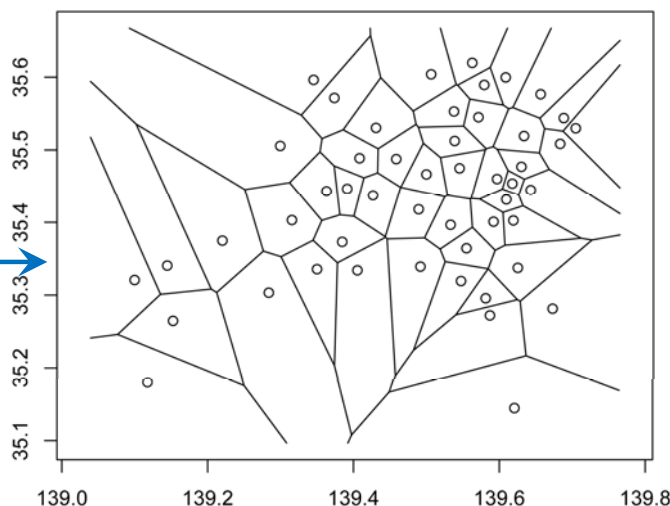
```
> plot(tl)
```

x座標として変数dfのlonを指定
y座標として変数dfのlatを指定

voronoi図だけを作り、変数tlに代入

変数tlをplot(描画)

※ボロノイ図【実線】
のみを描いた



変数 df の中身		df\$lon	df\$lat
name	type	lon	lat
鶴見区	ward	139.6825	35.5084
神奈川区	ward	139.6294	35.4769
西区	ward	139.6170	35.4536
中区	ward	139.6422	35.4447
...	...	...	...

「変数名\$項目名」

と書くと列のデータを意味する

例) df\$lon ... 変数dfの項目lonのデータ

例) df\$type ... 変数dfの項目typeのデータ

例) df\$name ... 変数dfの項目nameのデータ

RでVoronoi図を描く【csv編3】

- 母点のtype毎にVoronoi領域を色分けしたいとき

```
> df <- read.csv("Y:/R/kana.csv", header=T)
```

```
> df
```

変数dfの中身を確認

```
> dd <- deldir(df$lon, df$lat, z=df$type)
```

```
> tl <- tile.list(dd)
```

```
> plot(tl)
```

x座標として変数dfのlonを指定

y座標として変数dfのlatを指定

zとして変数dfのtypeを指定

変数dfの中身

name	type	lon	lat
鶴見区	ward	139.6825	35.5084
神奈川区	ward	139.6294	35.4769
西区	ward	139.6170	35.4536
中区	ward	139.6422	35.4447
...	...	...	...

df\$type

df\$lon

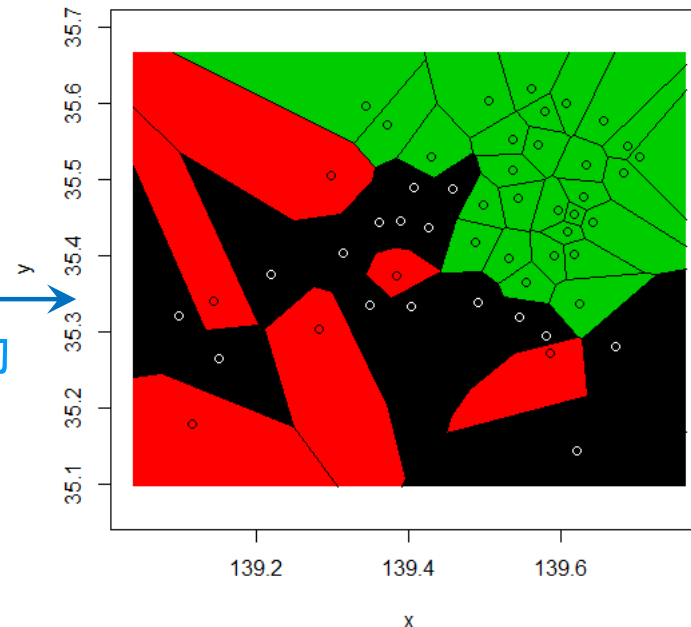
df\$lat

※ボロノイ図【実線】の
各母点のボロノイ領域を
type毎に色分けする
色は指定しなければ自動
例では、

ward (区) = 緑色

city (市) = 黒色

county (郡) = 赤色



注) county は主に米国の郡をさす
「高座郡寒川町」と言いたい時は
Samukawa Town, Koza District等

問題解決技法入門

2. GIS, Voronoi diagram

RでVoronoi図を描く【[google地図](#)編】

RでVoronoi図を描く【地図編1】

- Voronoi図とGoogle地図を重ねて描く
- 準備

```
> install.packages("deldir")  
> install.packages("ggmap")
```

packageのインストール
(deldir と ggmap)
※インストール済みなら不要

```
> library(deldir)  
> library(ggmap)
```

パッケージの読み込み
(deldir と ggmap)
※読み込み済みなら不要

— Tips! (裏技・小ネタ)

```
> library()
```

※現在インストールされているパッケージのリストが表示される

```
> setwd("C:/Users/〇〇/Desktop/R")
```

[ctrl]+[p] ...1つ前の既出命令
[ctrl]+[n] ...1つ後の既出命令
をそれぞれ表示

※同じ命令を少し修正して書きたい
場合などに便利

※wd(working directory)設定 設定しておくで、ここにあるファイルはファイル名だけで指定できる
※例はWindowsのデスクトップにRフォルダを作っている場合(〇〇にはユーザー名が入る)

RでVoronoi図を描く【地図編2】

- ファイル(*.csv)の読み込み

```
> df <- read.csv("Y:/R/kana.csv", header=T)
```

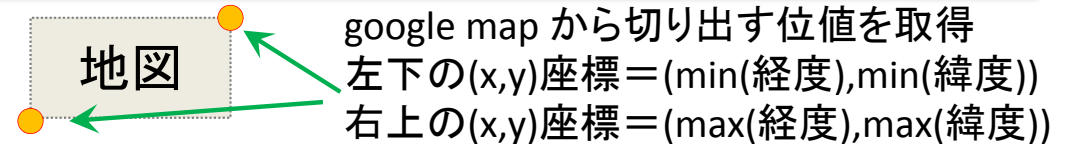
name	type	lon	lat
鶴見区	ward	139.6825	35.5084
神奈川区	ward	139.6294	35.4769
西区	ward	139.6170	35.4536
中区	ward	139.6422	35.4447
...	...	...	...

kana.csv

- 経度・緯度の最小値・最大値を取得

```
> loc <- c(min(df$lon), min(df$lat), max(df$lon), max(df$lat))
```

- Google map の取得



```
> GM <- ggmap(get_map(location=loc, zoom=10, source="google"))
```

※location ... 切り出す地図の位置(左下(x,y)座標-右上(x,y)座標の4点で指定)

※zoom ... 地図の縮尺指定: 1市が対象なら12ぐらい, 1県が対象なら10ぐらいに設定

※source ... 取得する地図のソース(源, 元, 情報源, 出所)

- 地図描画確認

```
> GM
```

※欲しいサイズになっているかどうか確認

縮尺はzoom で調整するので, 値を小さくしたり大きくしたりして再度読み込み直す

RでVoronoi図を描く【地図編3】

• 母点の描画準備

```
> GP <- geom_point(data=df, aes(x=lon, y=lat, colour=factor(type)),  
size=3)
```

母点の大きさを3に設定

dataとしてdfを使う

x座標としてdfのlon

y座標としてdfのlatを使う

母点の色分けにdf
のtypeを利用する

• ボロノイ領域の描画準備

```
> dd <- deldir(df$lon, df$lat)  
> GS <- geom_segment(data=dd$dirsgs, aes(x=x1, y=y1, xend=x2,  
yend=y2), size=0.5, linetype=1)
```

dataとして変数ddのdirsgsを使う

(x,y)座標としてdd\$dirsgsの(x1,y1)を
(xend,yend)としてdd\$dirsgsの(x2,y2)を使う

※線の太さ

※線種

実線 = 1

各種点線 = 2,3,4,5,...

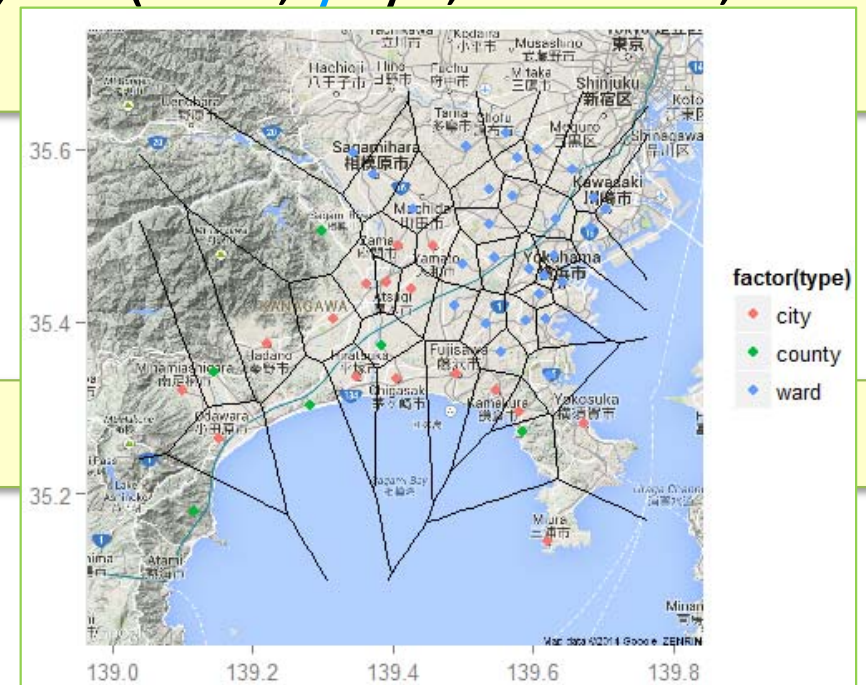
• 描画

```
> GM+GP+GS
```

※google map

※母点

※ボロノイ領域



問題解決技法入門

2. GIS, Voronoi diagram

RでVoronoi図を描く【地図番外編】

RでVoronoi図を描く【地図番外編1】

- ボロノイ領域を色分けするには...
- 準備

```
> install.packages("ggplot2", dependencies=TRUE)
```

packageのインストール
(ggplot2)

※地図編に追加して必要
※インストール済みなら不要

```
> library(ggplot2)
```

```
> library(reshape2)
```

```
> library(plyr)
```

パッケージの読み込み
(ggplot2, reshape2, plyr)

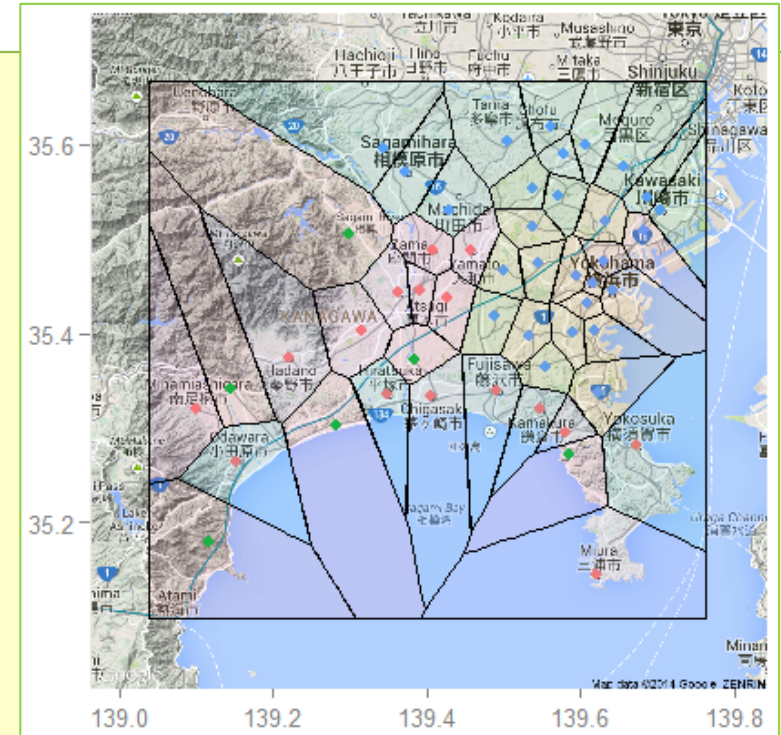
※地図編に追加して必要
※読み込み済みなら不要

RでVoronoi図を描く【地図番外編2】

※参考HP: Data Science by R and Python オープンデータを解析する
<http://tomoshige-n.hatenablog.com/entry/2014/08/15/002345>

• ボロノイ領域色分け

```
> tl <- tile.list(dd)
> uc <- unclass(tl)
> uc <- lapply(uc, function(l) {
+   data.frame(x=l$x, y=l$y)
+ })
> tldef <- melt(tl, id.vars=c("x"))
> tldef <- tldef[,c(4,1,3)]
> names(tldef) <- c('tile', 'lon', 'lat')
> GV <- geom_polygon(data=tldef, aes(x=x, y=y, fill=factor(tile)),
+ colour='black', alpha=0.1)
```



※ボロノイ領域の枠線の色

※ボロノイ領域の透過率 (alpha=0.0[透明]~ 1.0[無透過])

• 描画

```
> GM+GP+GV
```

※GMとGPは地図編で作成した「google map」と「母点」

もっと詳しく・参考文献

- もっと詳しく

- GIS, voronoi図を使った分析・研究をやりたい

- 根本研・堀田研

- 参考文献

- U.リゲス「Rの基礎とプログラミング技法」シュプリンガー(2006)
 - 岡田昌史編「The R Book」九天社(2004)
 - 金明哲「Rによるデータサイエンス」森北出版(2007)
 - 青木繁伸「Rによる統計解析」オーム社(2009)
 - 荒木孝治「RとRコマンダーではじめる多変量解析」日科技連(2007)
 - 長畑・大橋「Rで学ぶ経営工学の手法」共立出版(2008)
 - R情報サイト(ex. RjpWiki = <http://www.okada.jp.org/RWiki/>)
 - R for Windowsインストール(ex. <http://cran.md.tsukuba.ac.jp/bin/windows/>)

演習：ボロノイ図を用い地図上でのコンビニ勢力図を作る

1. データ取得①

- 「コンビニマップ」(<http://cvs-map.jp/>)の各担当データを取得(「conv.csv」にコピー&csvで保存)

no	area	name	address	type	lat	lon
1	横浜市鶴見区	セブンイレブン横浜上末吉3丁目店	上末吉3-12-7	セブンイレブン		
2	横浜市鶴見区	まいばすけっと上末吉5丁目店	上末吉5-19-1 メゾンMK	まいばすけっと		
3	横浜市鶴見区	セブンイレブン上末吉店	上末吉5-2-19	セブンイレブン		
4	横浜市鶴見区	ローソン鶴見上末吉五丁目店	上末吉5-7-3	ローソン		

2. データ取得②

- 「Geocoding - 住所から緯度経度を検索」(<http://www.geocoding.jp/>)で住所からlon,latを取得し, csvファイルにcopy&paste lat:緯度(世界測地系・北緯), lon:経度(世界測地系・東経)

no	area	name	address	type	lat	lon
1	横浜市鶴見区	セブンイレブン横浜上末吉3丁目店	上末吉3-12-7	セブンイレブン	35.4763	139.6064
2	横浜市鶴見区	まいばすけっと上末吉5丁目店	上末吉5-19-1 メゾンMK	まいばすけっと	35.47637	139.6046
3	横浜市鶴見区	セブンイレブン上末吉店	上末吉5-2-19	セブンイレブン	35.47675	139.6046
4	横浜市鶴見区	ローソン鶴見上末吉五丁目店	上末吉5-7-3	ローソン	35.47677	139.6144

3. Voronoi図を描く

- google地図上に, コンビニを母点としたボロノイ図を描き(地図編参照, 母点はtypeで色分け, google地図の縮尺(zoom)は適切な大きさに), 結果を図(pngファイル「conv.png」)として保存

4. データファイル(conv.csv)と描画図(conv.png)を提出