

第 5 章: 発見 (5.3. 空間データ)

今井耕介 著

『社会科学のためのデータ分析入門 (QSS)』

2026-03-09

5.3 空間データ

地理空間情報の分析

- ▶ **空間データ**: 位置情報（緯度・経度など）を持つデータ。
- ▶ **目標**: 現象の地理的な広がりを視覚化し、地域間のパターンや関係を理解する。
- ▶ **古典的な例**: スノーによるコレラ発生源の特定。

5.3.1 R における空間データの扱い

maps パッケージ: プロットのコード

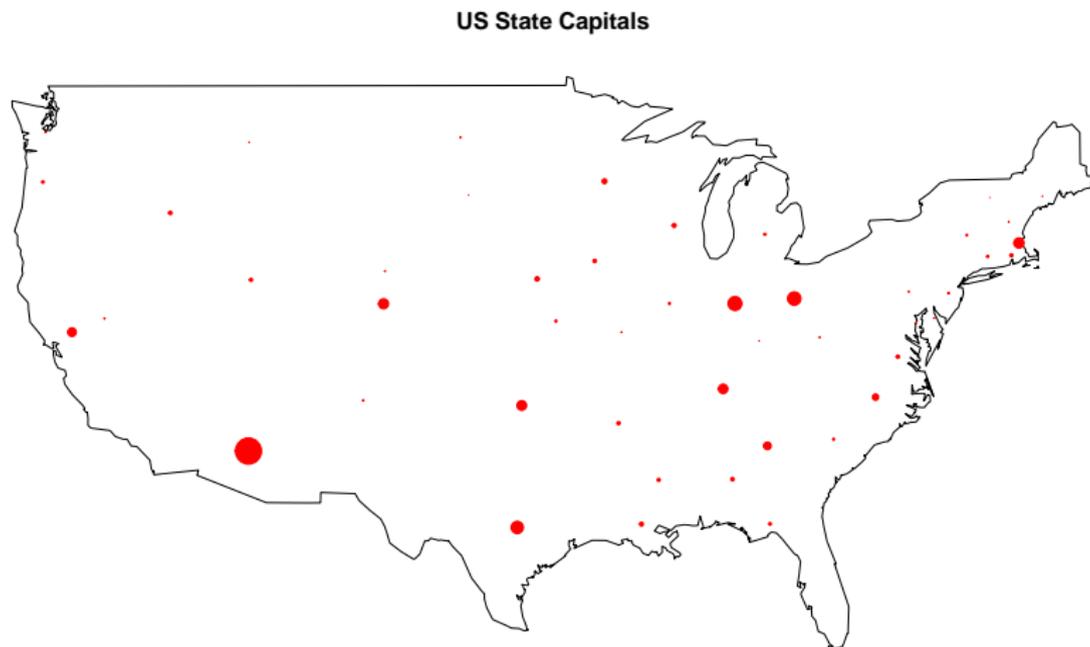
▶ R で地図を描画するための標準的なパッケージ。

```
library(maps)
# 1. 米国地図の描画
map(database = "usa")

# 2. 州都のデータ (us.cities) を読み込む
data(us.cities)
# capital == 2 が州都を表す
capitals <- subset(us.cities, capital == 2)

# 3. 州都の位置を地図上にプロット
# cex は点の大きさ、pch は点の形
points(x = capitals$long, y = capitals$lat,
       cex = capitals$pop / 500000, pch = 19, col = "red")
```

maps パッケージ: 描画結果



特定の地域の拡大: プロットのコード

▶ `regions` 引数で特定の州などを指定できます。

1. カリフォルニア州のみを描画

```
map(database = "state", regions = "California")
```

2. カリフォルニア州の都市データを抽出

```
cal.cities <- subset(us.cities, subset = (country.etc == "CA"))
```

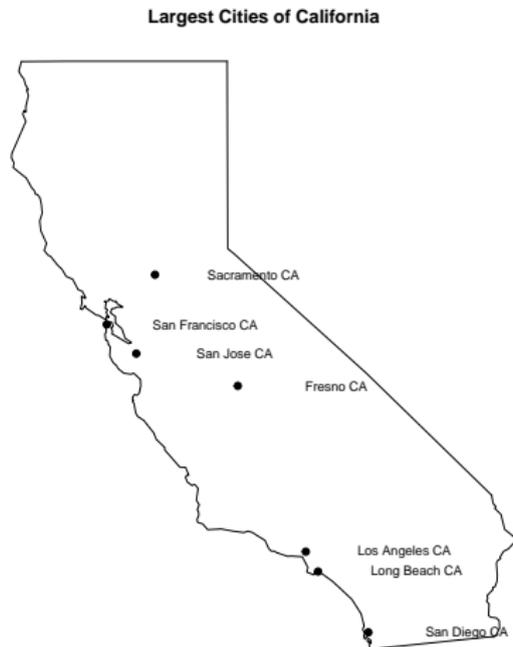
3. 人口上位 7 都市を特定し、プロット

```
top7 <- order(cal.cities$pop, decreasing = TRUE)[1:7]  
points(x = cal.cities$long[top7], y = cal.cities$lat[top7], pch = 19)
```

4. 都市名を追加

```
text(x = cal.cities$long[top7] + 2, y = cal.cities$lat[top7],  
     label = cal.cities$name[top7], cex = 0.8)
```

特定の地域の拡大: 描画結果



5.3.2 R における色の指定

rgb() 関数と透明度

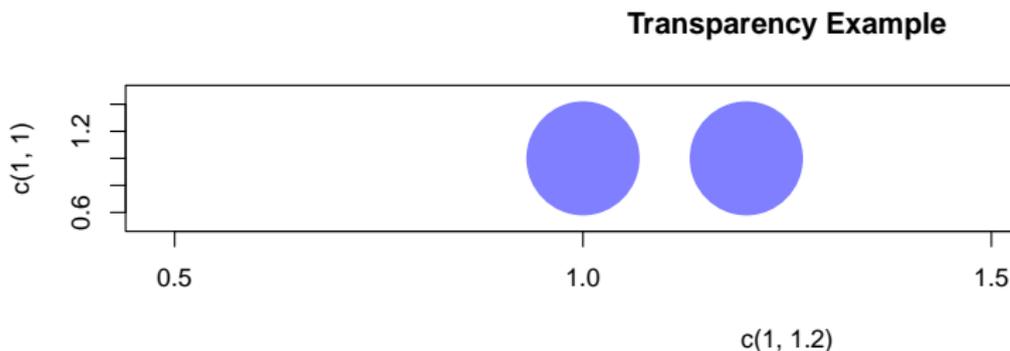
- ▶ `rgb()` で赤・緑・青の強度 (0~1) と、透明度 (`alpha`) を指定できます。

```
# 半透明の青 (alpha=0.5) を作成
```

```
blue.trans <- rgb(red = 0, green = 0, blue = 1, alpha = 0.5)
```

```
# 重なりを確認するためのプロット
```

```
plot(x = c(1, 1.2), y = c(1, 1), xlim = c(0.5, 2), ylim = c(0.5, 1.5),  
     pch = 16, cex = 10, col = blue.trans, main = "Transparency Example")
```



5.3.3 米国大統領選挙のマップ

pres08 データの読み込み (1)

- ▶ 民主党が勝った州と共和党が勝った州の名前を、それぞれ別々のリストにまとめます。

```
# 1. ローカルに保存したデータの読み込み (推奨)
```

```
pres08 <- read.csv("pres08.csv")
```

```
# (参考) URL から直接読み込むことも可能
```

```
# pres08 <- read.csv("https://ayumu-tanaka.github.io/QSS/QSS_Data/pres08.csv")
```

州のリストを作成 (2)

```
# 1. オバマ (民主党) が勝った州の名前を取り出す
# Obama > McCain という条件に合う state.name を抽出
blue_states <- pres08$state.name[pres08$Obama > pres08$McCain]

# 2. マケイン (共和党) が勝った州の名前を取り出す
# McCain > Obama という条件に合う state.name を抽出
red_states <- pres08$state.name[pres08$McCain > pres08$Obama]

# リストの最初の方を確認
head(blue_states, n = 5)

## [1] "California" "Colorado" "Connecticut" "D.C." "Delaware"
```

州別の色分け (2): 地図を塗るコード

- ▶ 作成したリストを使って、指定した州をまとめて塗りつぶします。

```
library(maps)
# 1. まず米国本土の白地図を描く
map(database = "state")

# 2. 「青い州のリスト」に含まれる州を青色 (blue) で一気に塗る
# regions 引数に名前のリストを渡す
map(database = "state", regions = blue_states,
     col = "blue", fill = TRUE, add = TRUE)

# 3. 「赤い州のリスト」に含まれる州を赤色 (red) で一気に塗る
map(database = "state", regions = red_states,
     col = "red", fill = TRUE, add = TRUE)
```


5.3.4 ウォルマートの拡大

walmart データの読み込み (1)

▶ 拡大プロセスの視覚化のため、出店データを読み込みます。

1. ローカルに保存したデータの読み込み (推奨)

```
walmart <- read.csv("walmart.csv")
```

(参考) URL から直接読み込むことも可能

```
# walmart <- read.csv("https://ayumu-tanaka.github.io/QSS/QSS_Data/walmart.csv")
```

拡大プロセスの視覚化 (2): 準備

```
# 1. 出店形態に応じた「色の列」をデータに追加する
walmart$col <- NA # 初期化

# 店舗タイプごとに、rgb() で透過色を指定
# Wal-MartStore は赤 (red)
walmart$col[walmart$type == "Wal-MartStore"] <- rgb(1, 0, 0, 0.2)
# SuperCenter は緑 (green)
walmart$col[walmart$type == "SuperCenter"] <- rgb(0, 1, 0, 0.2)
# DistributionCenter は青 (blue)
walmart$col[walmart$type == "DistributionCenter"] <- rgb(0, 0, 1, 0.5)
```

拡大プロセスの視覚化 (3): プロットのコード

```
# 1. 白地図を描く
```

```
map(database = "state")
```

```
# 2. 店舗の位置 (long, lat) を、設定した色 (col) でプロット
```

```
# cex (点の大きさ) は、配送センター (DistributionCenter) のみ大きくする
```

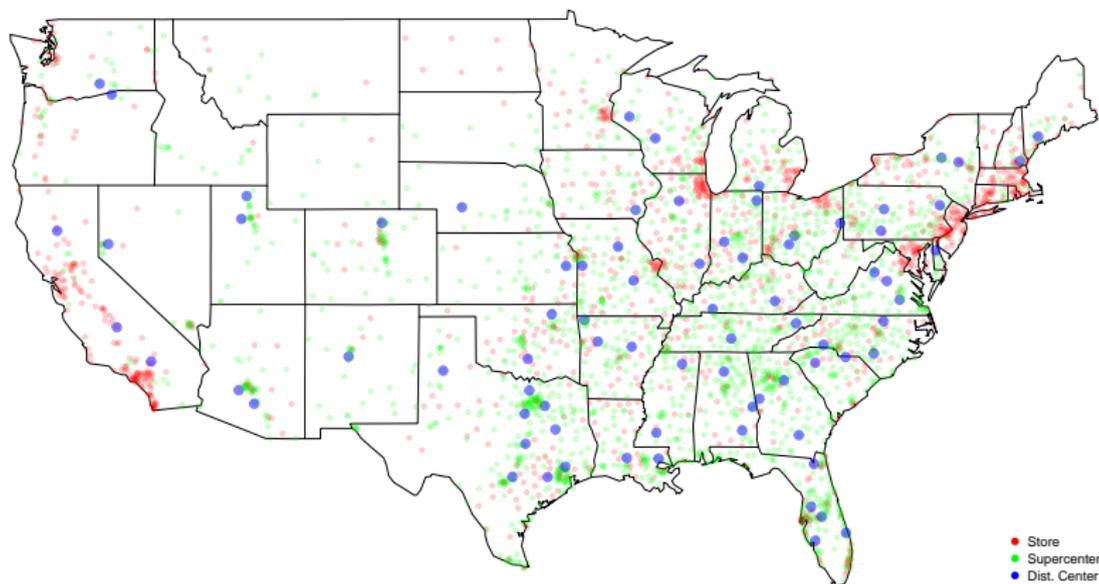
```
points(walmart$long, walmart$lat, col = walmart$col,  
       pch = 19, cex = ifelse(walmart$type == "DistributionCenter", 1, 0.5))
```

```
# 3. 凡例を追加
```

```
legend("bottomright", legend = c("Store", "Supercenter", "Dist. Center"),  
      col = c("red", "green", "blue"), pch = 19, bty = "n", cex = 0.7)
```

拡大プロセスの視覚化 (4): ウォルマートの拡大マップ

Expansion of Walmart



5.3.5 まとめ

まとめ

- ▶ **空間分析**: 現象の地理的パターンを視覚化する強力な手法。
- ▶ **maps パッケージ**: 基本的な地図描画と、特定地域の拡大が可能。
- ▶ **座標とプロット**: 緯度・経度を `points()` 等で地図上に重ねる。
- ▶ **色の活用**: `rgb()` と `alpha` を用いて、データの密度や属性を表現する。
- ▶ **コロプレスマップ**: 地域 (州など) を属性データに基づいて色分けする。