

第9回 基礎ゼミ

沖縄観光データ分析

福田研究室 学部4年
五百藏夏穂

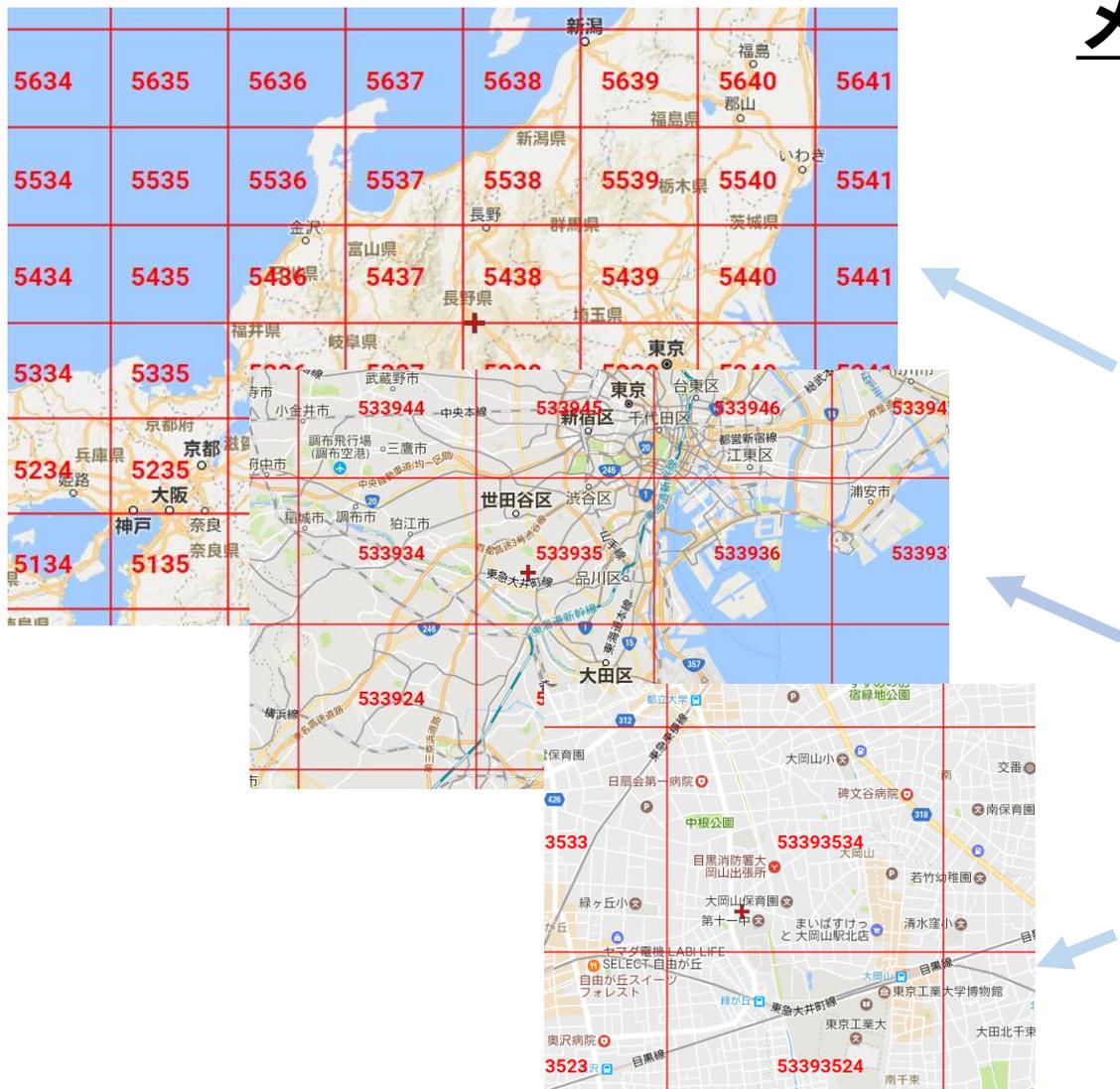
コンテンツ

- メッシュデータについて
- 沖縄での応用例
 - ① モバイル空間統計
 - ② 国土数値情報 宿泊容量データ
- 回帰分析

メッシュデータについて

メッシュデータ

地域を格子状に区切った単位で
その範囲の情報を表すのに利用される。



1次メッシュ 1度ごとの経線と2/3ごとの緯線で分割

80km×80km

ex)5339

2次メッシュ

1次メッシュを縦横8等分にしたもの

10km×10km

ex)533935

3次メッシュ

2次メッシュを縦横10等分にしたもの

1km×1km

ex)53393534

メッシュデータについて

パッケージのインストール

```
1 ##パッケージのインストール
2
3 install.packages("jpmesh", dependencies=TRUE)
4 install.packages("ggthemes", dependencies=TRUE)
5 install.packages("viridis", dependencies=TRUE)
6
7 library(jpmesh)
8 library(dplyr, warn.conflicts=FALSE)
9 library(ggplot2)
10 library(ggthemes)
11 library(mapproj)
12 library(knitr)
13 library(viridis)
14
15
```

“jpmesh” : 地域メッシュをRから
利用可能にする

“ggthemes” : 図示の際体裁を整える

“viridis” : 色分けして表示する

メッシュデータについて

Rでのメッシュコードの利用①

```
16
17 #メッシュコードからメッシュ範囲特定のための緯度経度の取得
18 meshcode_to_latlon(5133)
19 meshcode_to_latlon(513377)
20 meshcode_to_latlon(51337783)
21
22 #緯度経度から、範囲内のメッシュコードを取得
23 latlong_to_meshcode(34, 133, order = 1)
24 latlong_to_meshcode(34.583333, 133.875, order = 2)
25 latlong_to_meshcode(34.65, 133.9125, order = 3)
26
27 #Detect fine mesh code (より小さいメッシュコードの取得)
28 detect_mesh(52350422, lat = 34.684176, long = 135.526130)
29 detect_mesh(523504221, lat = 34.684028, long = 135.529506)
30
```

“jpmesh”

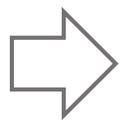
地域メッシュデータを利用可能に

- ・ 緯度、経度を出力
meshcode_to_latlon(メッシュコード番号)
- ・ メッシュコードを出力
latlong_to_miscode(緯度,経度,order=次元)
- ・ より高次元のメッシュコードを出力
detect_mesh(元のメッシュコード,
lat=緯度,long=経度)

沖縄での応用例

メッシュ内
滞在人口

メッシュ内
施設数



昼間滞在人口

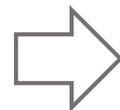
×

観光施設数

夜間滞在人口

×

宿泊容量



地図にプロットして相関を見る

沖縄での応用例

日時

メッシュ

滞在人口

	A	B	C	D	E	F	G	H
1	date	day_of_wetime		area	residence	age	gender	population
2	20160819	15	400	39270593	-1	-1	-1	12
3	20160819	15	400	39270594	-1	-1	-1	24
4	20160819	15	400	39271503	-1	-1	-1	14
5	20160819	15	400	39271504	-1	-1	-1	23
6	20160819	15	400	39271505	-1	-1	-1	19
7	20160819	15	400	39271506	-1	-1	-1	11
8	20160819	15	400	39271507	-1	-1	-1	25
9	20160819	15	400	39271512	-1	-1	-1	14
10	20160819	15	400	39271513	-1	-1	-1	29
11	20160819	15	400	39271514	-1	-1	-1	14
12	20160819	15	400	39271515	-1	-1	-1	25
13	20160819	15	400	39271516	-1	-1	-1	20
14	20160819	15	400	39271517	-1	-1	-1	25
15	20160819	15	400	39271522	-1	-1	-1	17
16	20160819	15	400	39271523	-1	-1	-1	21
17	20160819	15	400	39271524	-1	-1	-1	22

モバイル空間統計データ

NTT ドコモの基地局データ

3次元メッシュごとの
日時別滞在人数

沖縄での応用例

メッシュ 宿泊容量

1	メッシュ	宿泊者数
2	39270000	0
3	39270001	0
4	39270002	0
5	39270003	0
6	39270004	0
7	39270005	0
8	39270006	0
9	39270007	0
10	39270008	0
11	39270009	0
12	39270010	0
13	39270011	0
14	39270012	0
15	39270013	0
16	39270014	0

国土交通省 国土数値情報

宿泊容量メッシュデータ

3次メッシュごとの宿泊容量
(宿泊施設収容可能人数の総計)

沖縄での応用例

準備

```
49
50 ▾ #####
51
52
53 # モバイル空間統計データ読み込み
54 mobile_stat <- read.csv("201608_DAY_JP_1km_Pop.csv")
55 mobile20160827400 <- subset(mobile_stat, date==20160826 & time==400)
56
57
58 #write.csv(mobile20160827400,"mobile400.csv")
59
60 # 沖縄本島市区町村コード
61 honto_list <- c(47201, 47205, 47208, 47209, 47210, 47211, 47212,
62               47213, 47215, 47301, 47302, 47303, 47306, 47308,
63               47311, 47313, 47314, 47315, 47324, 47325, 47326,
64               47327, 47328, 47329)
65
66
67 # 沖縄本島のメッシュだけを抽出
68 honto_mesh0 <- subset(pref_mesh(47), city_code %in% honto_list)
69
70
71 ▾ #####
72
```

地図の中身(data)の準備

- ・ 特定のデータのみ抽出

subset(データ,
date==日付 かつ time==時間)

地図の枠組み(map)の準備

- ・ 本島の市区町村コードだけを行列にする

- ・ 市町村コードとの重なりから
データを本島にしぼる

subset(データ,city_code %in% honto_list)
A %in% B : AとBの重なるもの

沖縄での応用例

枠組み (map) の作成

```
70
71 ▾ #####
72
73
74 # メッシュの枠を作成
75 df.map <- honto_mesh0 %>%
76   mutate(mesh_area = purrr::map(id, meshcode_to_latlon)) %>%
77   tidyr::unnest() %>%
78   mutate{lng1 = long_center - long_error,
79          lat1 = lat_center - lat_error,
80          lng2 = long_center + long_error,
81          lat2 = lat_center + lat_error)
82
83 ggplot() +
84   geom_map(data = df.map,
85           map = df.map,
86           aes(x = long, y = lat, map_id = id),
87           fill = "white", color = "black") +
88   coord_map(projection = "mercator") -> plot_map
89
90 # メッシュの枠を表示
91 plot_map
92
93 ▾ #####
94
```

“dplyr”

データフレームの操作

A %>% B

Aを因数としてBに渡す

mutate : 列を加える

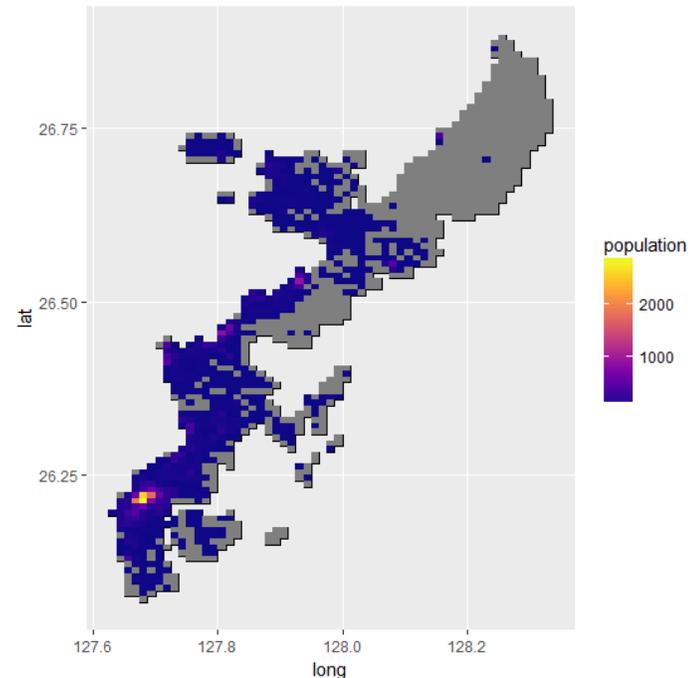
tidyr : データ変形(データフレームを展開)

“ggplot2”

沖縄での応用例

ヒートマップ表示(モバ空)

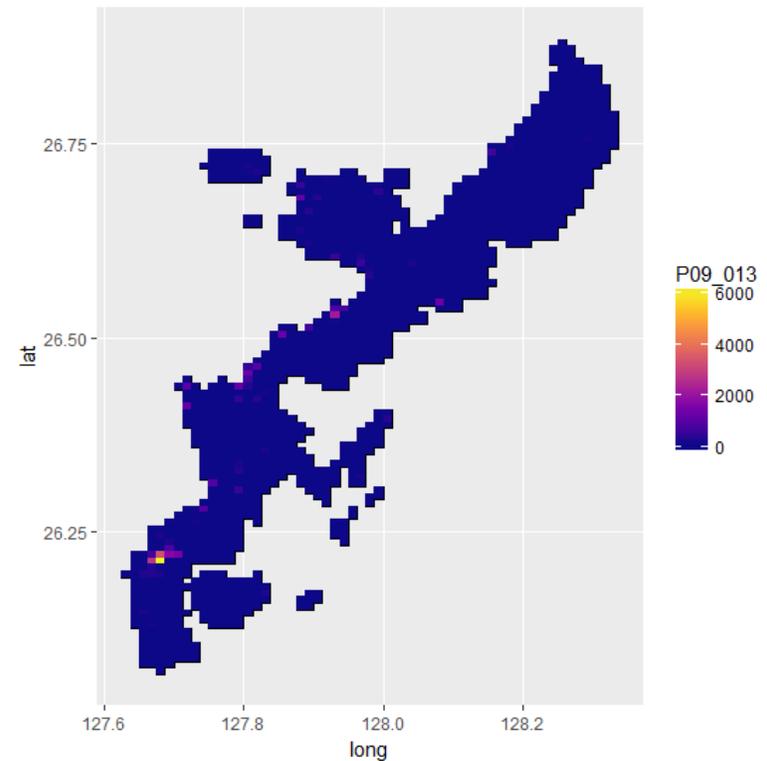
```
92
93 ▾ #####
94
95
96 # モバ空データのヒートマップ表示
97
98 df.map.mobile <- merge(df.map, mobile20160827400,
99                       by.x = "id", by.y = "area", all.x = T)
100
101 plot_map +
102   geom_map(data = df.map.mobile,
103           map = df.map,
104           aes(fill = population, map_id = id, stat = "identity"))
105   scale_fill_viridis(option="plasma")
106
107
108 ▾ #####
109
```



沖縄での応用例

ヒートマップ表示(宿泊)

```
107
108 ▾ #####
109
110
111 #宿泊者数データ
112
113 syukuhaku <- read.csv("syukuhaku.csv")
114
115 #メッシュの枠は上記のものと同じ
116
117 #宿泊者数データのヒートマップ表示
118
119 df.map.syukuhaku <- merge(df.map, syukuhaku,
120                          by.x = "id", by.y = "P09_001", all.x
121
122 plot_map +
123   geom_map(data = df.map.syukuhaku,
124           map = df.map,
125           aes(fill = P09_013, map_id = id, stat = "identity"
126             scale_fill_viridis(option="plasma")
127
128 ▾ #####
129
```

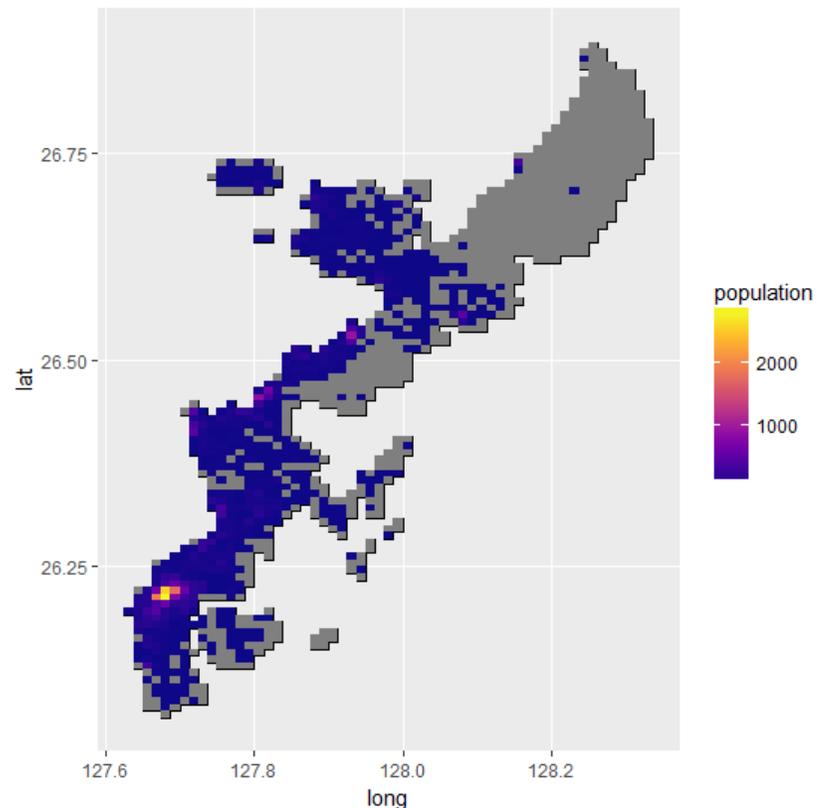


沖縄での応用例

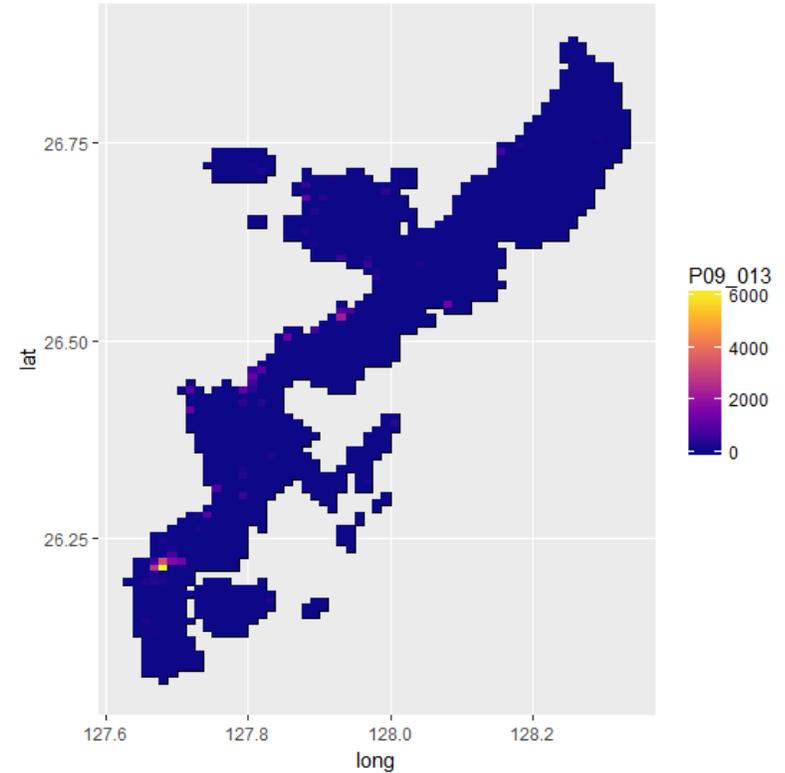
視覚化で相関がありそう...! ?



回帰分析



夜間滞在人口



宿泊容量

回帰分析

回帰分析

```
128 #####  
129  
130  
131 #回帰分析  
132  
133  
134 match<-read.csv("match400.csv")  
135  
136 result<-lm(population ~ syukuhaku, data = match)  
137  
138  
139 summary(result)  
140  
141 plot(match$syukuhaku,match$population,xlab="宿泊容量(人)",ylab="時間人口(人)")  
142 abline(result)  
143
```

既に夜間人口と宿泊容量を同じデータ
メッシュごとに合わせてあるデータ

データCに属する
AとBの回帰分析

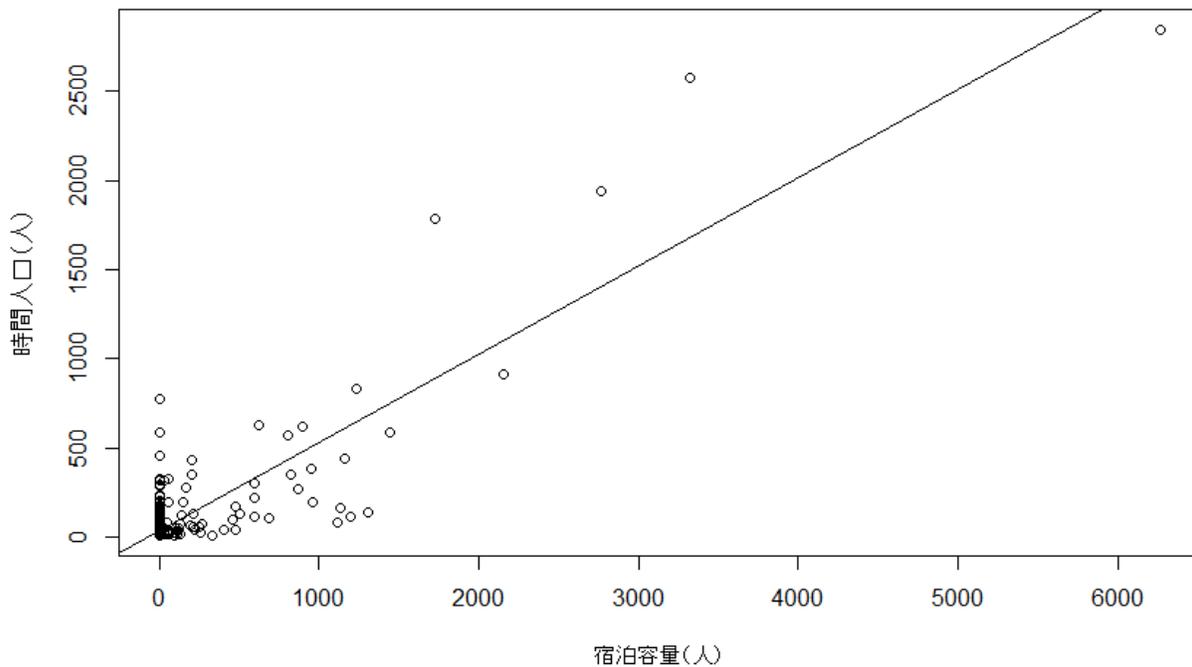
$\text{lm}(A \sim B, \text{data} = C)$

回帰分析の結果を表示

結果を散布図に図示
回帰直線を引く

回帰分析

回帰分析結果



説明変数(Y) : 宿泊容量

目的関数(X) : 滞在人口

$$Y=0.496X+33.27$$

t値も十分大きいので相関がある

```
> summary(result)
```

```
Call:
```

```
lm(formula = population ~ syukuhaku, data = match)
```

```
Residuals:
```

Min	1Q	Median	3Q	Max
-548.76	-21.27	-15.27	3.73	898.71

```
Coefficients:
```

	Estimate	Std. Error	t value	Pr(> t)
(Intercept)	33.27082	3.71372	8.959	<2e-16 ***
syukuhaku	0.49579	0.01054	47.044	<2e-16 ***

```
---  
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```
Residual standard error: 95.69 on 681 degrees of freedom  
Multiple R-squared:  0.7647,    Adjusted R-squared:  0.7643  
F-statistic: 2213 on 1 and 681 DF,  p-value: < 2.2e-16
```