Rを使った基礎集計
と
QGISによる可視化

Start-up Seminar #1

2020/04/07

M1 小関玲奈

Start-up Seminar 2020 Syllabus

■ 目標

推定を一通り実装する経験を得て，調べながらならコードを書けるようになる

■ 内容

- 毎回，小課題として，簡単な実装やコードの穴埋など，手を動かす課題を出す
- 第5回目にて，大課題の発表会(考察まで)を実施.

■ 大課題

PP調査による行動データに基づき，(Rを用いて)MNLによるパラメータ推定を行う.

また，推定したモデルで政策シミュレーションを行う.

■ 日程

#1 4/7 (Tue) 8:30～10:00
Rによる基礎集計，GIS

#2 4/14 (Tue) 8:30～10:00
MNLでの推定

#3 4/21 (Tue) 8:30～10:00
Pythonによる最短経路探索

#4 4/28 (Tue) 8:30～10:00
均衡配分

#5 5/12 (Tue) 8:30～10:00
課題発表

#6 5/19 (Tue) 8:30～10:00
経路選択モデル概論

#7 5/26 (Tue) 8:30～10:00
(予備)

今日の内容

～今日の目標～
Rに慣れること

1. はじめに
   - データの概要, 分析の大まかな流れ
   - 基礎集計とは
   - Excel による基礎集計
2. Rに慣れよう
   - Rのインストール
   - Rで計算する, 関数定義とプログラミング入門, グラフ作成入門
3. Rでデータ処理をしよう
   - 基本操作
   - Location データを編集してみる
   - Rでピボットテーブル, ピボットグラフ
4. 本日の小課題

はじめに

はじめに

■行動モデル

人の行動データから,

人の意思決定のメカニズムをモデル化する.

→どうしたら歩行者が増える？/ どうしたらみんなが避難できる？

■行動データ

・パーソントリップデータ(PT)

・プローブパーソンデータ(PP)

・Wi-fiデータ

・アンケート調査データなどなど

今回はPPデータについて扱います.

PPデータは主に，location data と trip data の2つ.

PPデータ

GPS機能を搭載した携帯電話と移動通信機器と 連動したWebダイアリーを用いて，モニタの移動活動記録と数秒間隔の位置情報を取得できる

✓ 大量，詳細な移動データ

✓ day-to-dayの行動記録

(同一個人の複数日にわたる行動履歴)

PTデータについて

■ trip data：tripごとのデータ

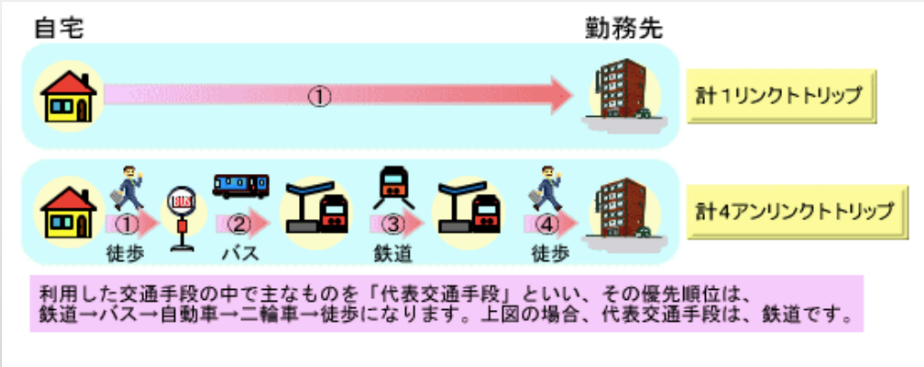
- tripID, userID, 移動目的, 移動手段, 出発・到着時刻, 出発地・到着地位置座標 など...

	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P
1	トリップID	ユーザーコー	目的コード	目的	出発日付	到着日付	トリップ時間	出発地コード	出発地	出発地ユーザ	出発地施設属	出発地施設属	出発地施設緯	出発地施設経	出発地施設緯	出発地施設経
2	15521	ec003	410	食事	2007/2/19 18:03	2007/2/19 18:41	2314	10578	松山河川国道	松山河川国道	120	勤務・通学・	33.8183764	132.742538	33.8216794	132.739998
3	15556	ec003	200	帰宅	2007/2/19 21:46	2007/2/19 22:14	1648	10581	なが坂	なが坂	190	飲食店・喫茶	33.8365256	132.772351	33.8398272	132.769808
4	15595	ec003	100	出勤・登校	2007/2/20 7:51	2007/2/20 8:19	1635	10579	自宅	自宅	110	自宅	33.8289447	132.783832	33.8322473	132.781288
5	15880	ec003	0	--	2007/2/20 21:03	2007/2/20 21:03	0									
6	15882	ec003	100	出勤・登校	2007/2/21 7:46	2007/2/21 8:11	1459	10579	自宅	自宅	110	自宅	33.8289447	132.783832	33.8322473	132.781288
7	15944	ec003	310	業務	2007/2/21 11:21	2007/2/21 12:57	5770	10578	松山河川国道	松山河川国道	120	勤務・通学・	33.8183764	132.742538	33.8216794	132.739998
8	16008	ec003	300	帰社・帰校	2007/2/21 15:58	2007/2/21 17:04	3958	10580	今治商工会議	今治商工会議	270	病院・医療施	34.0604778	133.00018	34.0637595	132.997608
9	16068	ec003	200	帰宅	2007/2/21 19:19	2007/2/21 19:40	1258	10578	松山河川国道	松山河川国道	120	勤務・通学・	33.8183764	132.742538	33.8216794	132.739998
10	16135	ec003	100	出勤・登校	2007/2/22 7:42	2007/2/22 8:02	1176	10579	自宅	自宅	110	自宅	33.8289447	132.783832	33.8322473	132.781288
11	16199	ec003	310	業務	2007/2/22 11:29	2007/2/22 12:21	3098	10578	松山河川国道	松山河川国道	120	勤務・通学・	33.8183764	132.742538	33.8216794	132.739998
12	16290	ec003	300	帰社・帰校	2007/2/22 16:27	2007/2/22 18:00	5573	12072	西条国道維持	西条出張所	130	事務所・会社	33.9087147	133.20576	33.9120162	133.203178
13	16406	ec003	100	出勤・登校	2007/2/23 7:38	2007/2/23 8:02	1455	10579	自宅	自宅	110	自宅	33.8289447	132.783832	33.8322473	132.781288

- tripとは

http://www.mlit.go.jp/crd/tosiko/pt.html

国土交通省-PT調査とは?



PPデータについて

■location data：各時点での位置情報のデータ(5~10秒間隔)

- tripID, locationID, userID, 移動手段,

時刻, 位置座標, 測位モード

- 例えば「鉄道で訪れた人の回遊範囲」をプロットするには？

移動手段が鉄道で, 鉄道＞徒歩（来街）・徒歩＞鉄道（帰宅）の転換があるもの…などの条件によって,

膨大なデータの中から適切なものを抽出する必要.

（その他マップマッチングなども. こちらは追い追い）

鉄道で訪れた人の回遊範囲

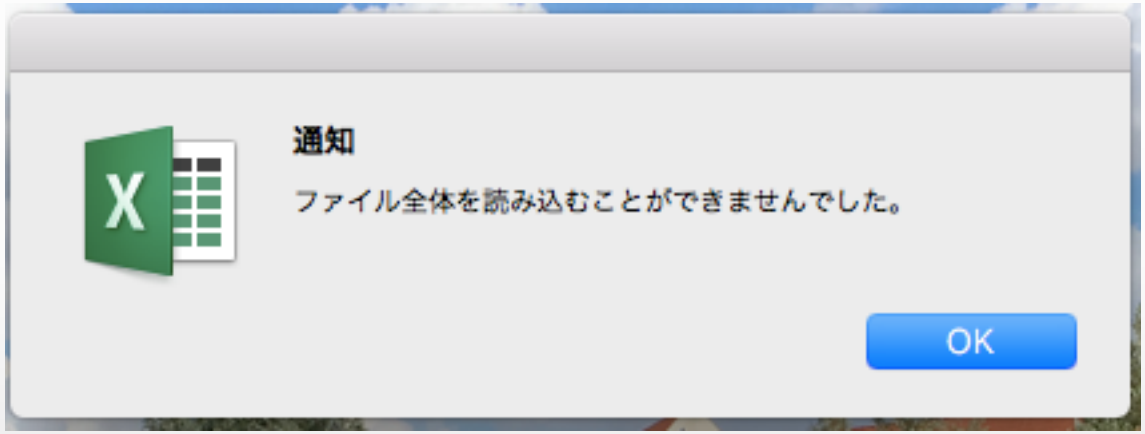
	A	B	C	D	E	F	G	H	I	J	K
1	トリップID	測位データID	移動手段	ユーザーコード	測位日時	緯度（日本測	経度（日本測	緯度（世界測	経度（世界測	測位モード	有効性
2	15375	2023091	バイク	ec029	2007/2/19 1:14	33.8348734	132.757748	33.8381749	132.755206	4	1
3	15375	2023092	バイク	ec029	2007/2/19 1:15	33.8304471	132.768558	33.8337493	132.766015	4	1
4	15375	2023093	バイク	ec029	2007/2/19 1:15	33.8350557	132.763462	33.8383573	132.760919	4	1
5	15375	2023094	バイク	ec029	2007/2/19 1:15	33.8348359	132.757517	33.8381374	132.754975	4	1
6	15375	2023095	バイク	ec029	2007/2/19 1:15	33.8305974	132.768429	33.8338996	132.765886	4	1
7	15375	2023096	バイク	ec029	2007/2/19 1:16	33.8348734	132.757748	33.8381749	132.755206	4	1

PPデータについて

■location data：各時点での位置情報のデータ(5~10秒間隔)

- しかし，locatin dataはデータが大きすぎてExcelでは

開けない…→だから，Rなどのツールが必要．



	A	B	C	D	E	F	G	H	I	J	K
1	トリップID	測位データID	移動手段	ユーザーコード	測位日時	緯度（日本測	経度（日本測	緯度（世界測	経度（世界測	測位モード	有効性
2	15375	2023091	バイク	ec029	2007/2/19 1:14	33.8348734	132.757748	33.8381749	132.755206	4	1
3	15375	2023092	バイク	ec029	2007/2/19 1:15	33.8304471	132.768558	33.8337493	132.766015	4	1
4	15375	2023093	バイク	ec029	2007/2/19 1:15	33.8350557	132.763462	33.8383573	132.760919	4	1
5	15375	2023094	バイク	ec029	2007/2/19 1:15	33.8348359	132.757517	33.8381374	132.754975	4	1
6	15375	2023095	バイク	ec029	2007/2/19 1:15	33.8305974	132.768429	33.8338996	132.765886	4	1
7	15375	2023096	バイク	ec029	2007/2/19 1:16	33.8348734	132.757748	33.8381749	132.755206	4	1

分析の流れ

- **データの特性を知る**
  分析を行う前に，そのデータはどのような情報を取得したものなのか，データから何が分かるのか を把握しておきましょう.

- **基礎集計を行い，傾向を探る**
  様々な属性を掛け合わせて何と何に相関があるのか，その行動の要因となっているものは何か，仮説を立てます.

- **モデルを構築し，推定を行う**
  基礎集計から仮説がたったら，モデルを構築して因果関係を定量的に分析します.その選択行動に 効ている要因とは何かを把握します.

基礎集計とは

■ データ分析をする前にする下ごしらえ的なもの

・全体の傾向をみる
・データの分布をみる
・異常値を発見する
・欠損値を発見する
・モデルに入れる変数の検討をつける などなど…

■ 集計に使うソフトウェア

1. データ整理/正規化
2. データ集計　　R/Python/Java/Excel…

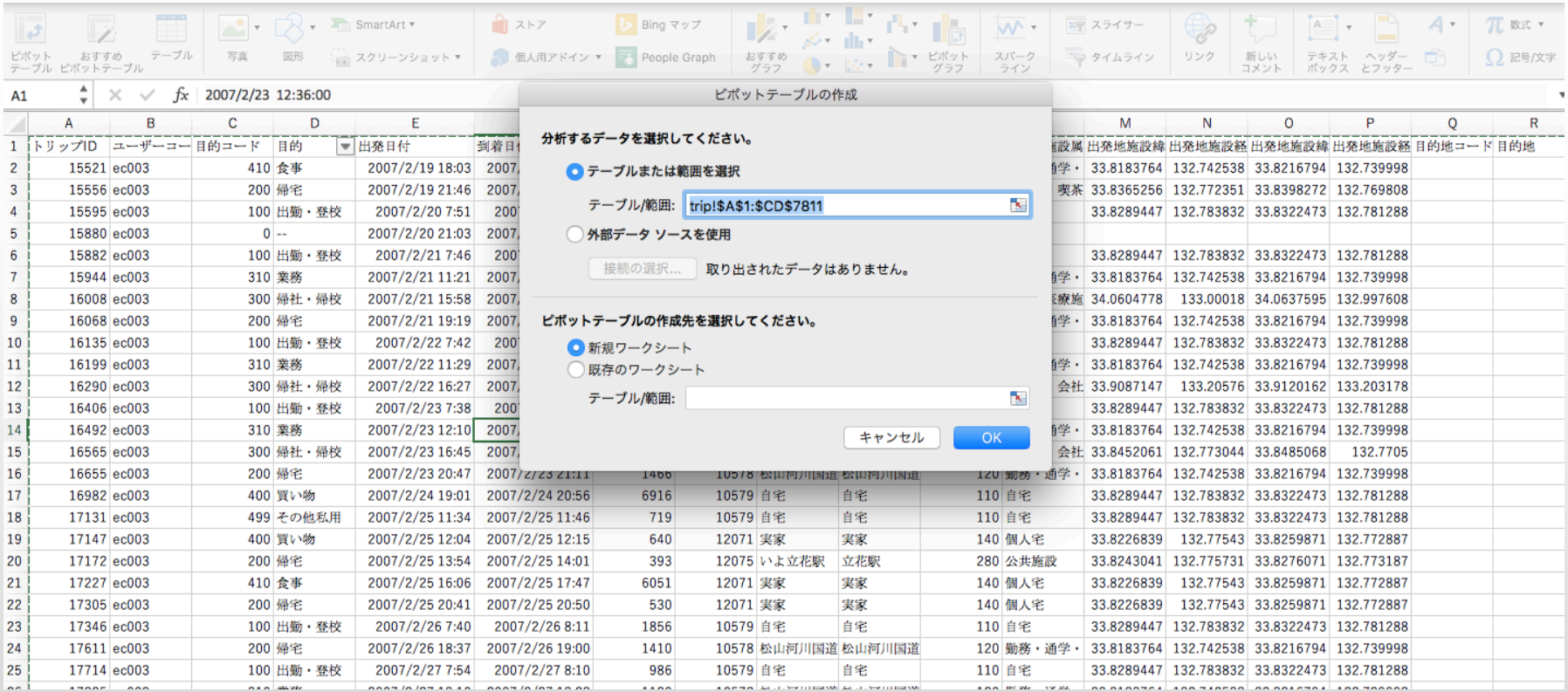
3. 可視化　　R/GIS/Excel/Google Earth…

Excelを使った基礎集計：ピボットテーブル

■ Excel：少量データを手軽にデータ集計，グラフ化，可視化が行える．

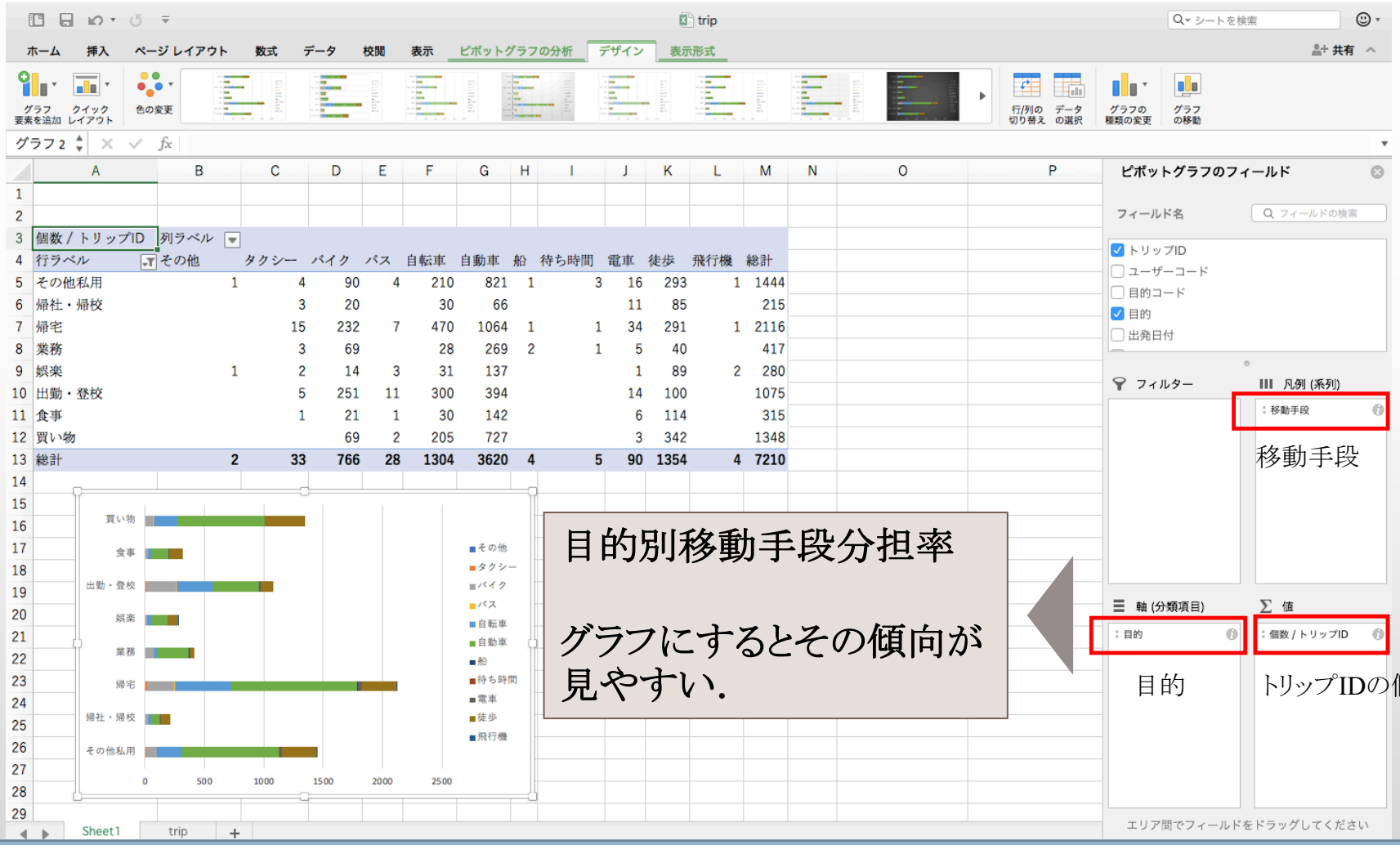
＞まずはtripデータをExcelのピボットテーブルで分析してみよう！

- [挿入]→ピボットテーブル→範囲を指定．



Excelを使った基礎集計：ピボットテーブル

- EX) 移動手段と移動目的の関係性を調べる.



個数 / トリップID	列ラベル											
行ラベル	その他	タクシー	バイク	バス	自転車	自動車	船	待ち時間	電車	徒歩	飛行機	総計
その他私用	1	4	90	4	210	821	1	3	16	293	1	1444
帰社・帰校		3	20		30	66			11	85		215
帰宅		15	232	7	470	1064	1	1	34	291	1	2116
業務		3	69		28	269	2	1	5	40		417
娯楽	1	2	14	3	31	137			1	89	2	280
出勤・登校		5	251	11	300	394			14	100		1075
食事		1	21	1	30	142			6	114		315
買い物			69	2	205	727			3	342		1348
総計	2	33	766	28	1304	3620	4	5	90	1354	4	7210

目的別移動手段分担率

グラフにするとその傾向が見やすい.

移動手段

目的

トリップIDの個数

Rに慣れよう

Rの導入

■S言語をオープンソース化したインタープリタ型言語

■統計処理に特化したプログラミング言語

■特徴
- ベクトル，行列演算が簡単
- そこそこ速い(らしい)
- 可視化もできる
- 初心者に優しい(でも他人の書いたコードが読みづらい)

■Tipsがネットに豊富

- **R-Tips**

http://cse.naro.affrc.go.jp/takezawa/r-tips/r.html
Rの基本操作はここにだいたい書いてある

- **RjpWiki**

http://www.okadajp.org/RWiki/

Rについてのwiki.みんなで編集できる

Rのインストール

RとRstudioをインストールしてください.

MacにRとRstudioをインストール

https://qiita.com/azzeten/items/1031c788ed093d3b3946

Windowsも調べれば出てきます.

■ R本体
https://cran.r-project.org/bin/macosx/

■ R studio Rの総合開発環境(IDE).他にもあるがこれがおすすめ.
https://www.rstudio.com/products/rstudio/download/

FreeのRStudio DesktopでOK

計算機として

```
> 1 + 2
[1] 3
>
> #これはコメントです．メモに便利：）
> #半角スペースはいくつあっても大丈夫だが，全角スペースは使わない．
>
> ( cos(0) - sqrt(2) ) / exp(4)
[1] -0.007586586
>
> #複雑な計算も簡単に
>
> x <- sqrt(2)
>
> #変数 <- 数値や計算式など
> #これで計算結果が変数に保存される．
>
> y <- 3
> x + y #変数同士の計算もできる
[1] 4.414214
```

ベクトルを扱う

```
> x <- c(1, 2, 3, 4, 5) #ベクトルを変数に代入
> sqrt(x) #xには数値だけでなくベクトルを入れることができる.
[1] 1.000000 1.414214 1.732051 2.000000 2.236068
>
> #ベクトルの各要素に対して演算を行う関数もたくさん
> #例えば, cor()で相関係数, mean()で平均値, sd()で標準偏差, summary()で要約統計量などなど
>
> x[2] #2番目の要素を取り出す.
[1] 2
>
> x[4]<-0  #4番目の要素を0に変更する
> x #xの値を確認する.
[1] 1 2 3 0 5
>
> y<-c(6, 7, 8)
>
> c(x, y) #ベクトルxとベクトルyの結合
[1] 1 2 3 0 5 6 7 8
>
```

関数定義とプログラミング入門

条件分岐：if, else

if(条件式) {
条件式がTRUEのときに実行される式
} else {
条件式がFALSEのときに実行される式
}

比較演算子

==	等しい
!=	等しくない
>=, <=	≥, ≤
>, <	>, <

複数の条件を指定するとき(値同士)

!	NOT（でない）
&&	AND(かつ)
||	OR（または）

ex: if ((-2<x) && (x<2))
→（xが-2より大）かつ（xが2未満）

```
mydistance <- function(a, b){
  if (a >= b){ #aがb以上ならば条件式はTRUEとなる
    return(a-b) #条件式がTRUEならばa-bが値として返される
  }
    else{ #aがbより小さければ条件式はFALSEとなる
      return(b-a) #条件式がFALSEならばb-aが値として返される
    }
}
```

```
> mydistance(5, 2)
[1] 3
>
> mydistance(2, 5)
[1] 3
> 4
```

```
myfunc <-function(x) {
  if (x < 0) return(-x)
  else if ((0 <= x) && (x < 5)) return(5)
  else return(x)
}
```

←3つ以上の
条件分岐をする際には
else if を用いる.

```
> myfunc(3)
[1] 5
```

関数定義とプログラミング入門

繰り返し文：for
for (ループ in リスト) {
繰り返す式…ループ変数がリストの範囲内である限り式が繰り返される
}

複数の条件を指定するとき(値同士)
! NOT（でない）
&& AND(かつ)
|| OR（または）

複数の条件を指定するとき(ベクトル同士)
! NOT（でない）
& AND(かつ)
| OR（または）

```
mysum <- function(n) { # 整数nを入れると1からnまでの和を返す関数を定義.
  i <- 0
  for (j in 1:n) {     # jが1からnまでの間...
    i <- i+ j          #「iにjを足す」を繰り返す
  }
  return(i)            # iの値を返す
}

mysum(5)
```

```
> mysum(5)
[1] 15
```

グラフ作成入門

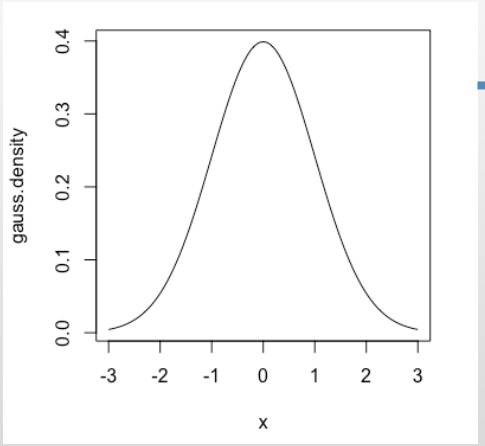
1. **高水準作図関数：1枚の完成されたグラフを描く**
   散布図：plot( ), ヒストグラム：hist( ), 棒グラフ：barplot( ) , 円グラフ：pie( )
   1次元関数のグラフ：curve(関数, 左端, 右端), 2次元関数のグラフ　persp(x軸, y軸, z軸) など…

2. 低水準作図関数：完成されたグラフに図形や文字などを追記する

```
gauss.density <- function(x){
  1 / sqrt(2 * pi) * exp(-x^2 / 2)
}
plot(gauss.density, -3, 3)
```

1. 1枚の完成されたグラフを描く
(1) プロットする関数を定義する

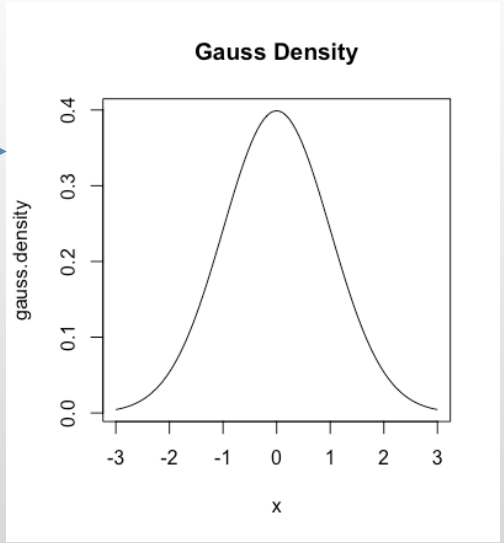
(2) 関数名を指定してプロット
plot(関数名, xの範囲の下限, 上限)



タイトル文字の追加

```
plot(gauss.density, -3, 3, main="Gauss Density")
```

(3) タイトルとラベルの制御, 軸の種類（対数軸など）, プロット点の形式, プロットの色や記号などを指定できる. 必要に応じて調べましょう.



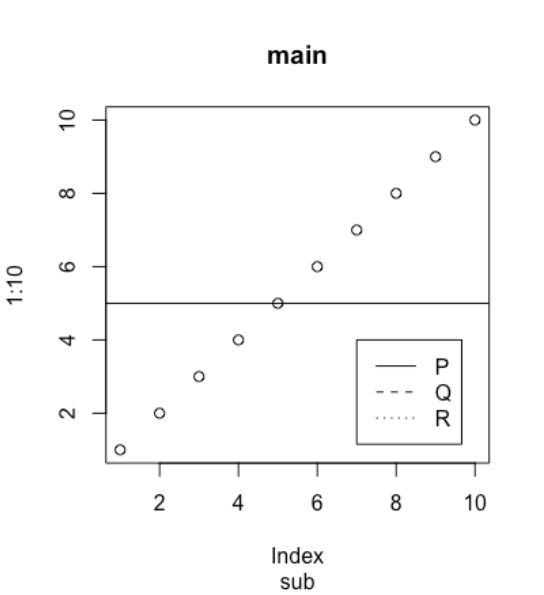
グラフ作成入門

1. **高水準作図関数：1枚の完成されたグラフを描く**
   散布図：plot( ), ヒストグラム：hist( ), 棒グラフ：barplot( ) , 円グラフ：pie( )
   1次元関数のグラフ：curve(関数, 左端, 右端), 2次元関数のグラフ　persp(x軸, y軸, z軸) など…

2. 低水準作図関数：完成されたグラフに図形や文字などを追記する

2. グラフを装飾
完成されたグラフに図形や文字などを追記するための関数.

```
plot(1:10) #散布図を描く
abline(h=5) #直線：y=5を追記
rect(1,6 4,9) #(1,6)から(4,9)に四角を追記
arrows(1,1 4,4) #(1,1)から(4,4)に矢印を追記
text(8,9 "text")
title("main", "sub")
legend(7, 4, lty=1:3,
       c("P", "Q", "R")) #右下に凡例を追記，ltyは線の種類
```

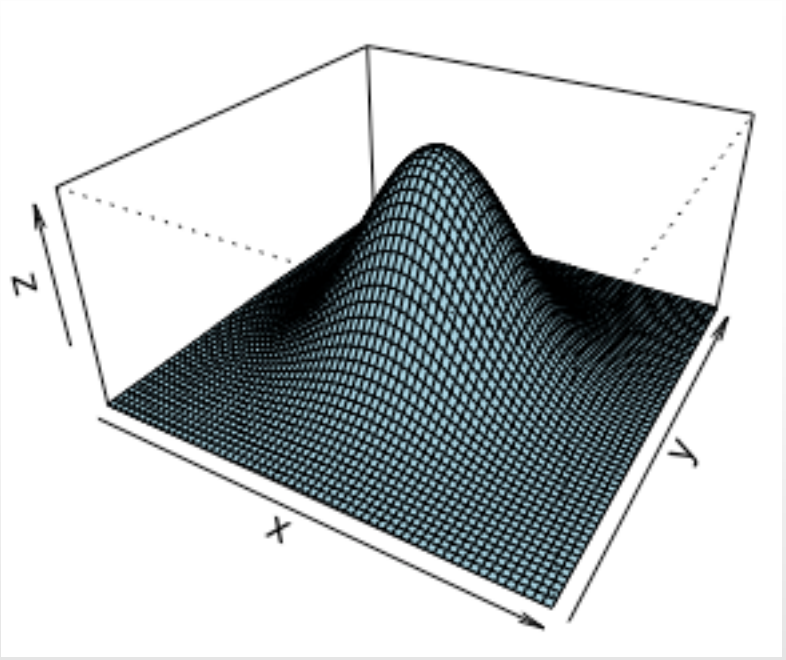


たくさんオプションがあります.
必要に応じて調べてみましょう.

グラフ作成入門

3次元も簡単に描けます．
使うときに調べてみてください．

```
x <- seq(-3, 3, length=61)
y <- x
f <- function(x, y){
  return( 1 / (2 *pi ) * exp(-(x^2 + y^2) / 2))
}

z <- outer(x, y, f)
persp(x, y, z, theta=30, phi=30, expand=0.5, col="lightblue")

}
```



Rでデータ処理を
してみよう

はじめに

データフレームとは

- 数値ベクトルや文字ベクトル，因子ベクトル（文字型ベクトル）などの異なる型のデータをまとめて1 つの変数として持っている**データフレーム(data.frame)**という型．各行と列にラベルが付いていて，ラベルによる操作ができる．統計処理がやりやすい型．

データを読み込む

- csvデータなどを読み込んで分析していく
- 作業ディレクトリの確認・データのあるディレクトリへの移動
  作業ディレクトリ：ファイルからデータやプログラムを読み込んだり，ファイルにデータを書き出したりする場所

現在の作業ディレクトリの確認
→getwd()

作業ディレクトリの変更
→setwd(“ “)

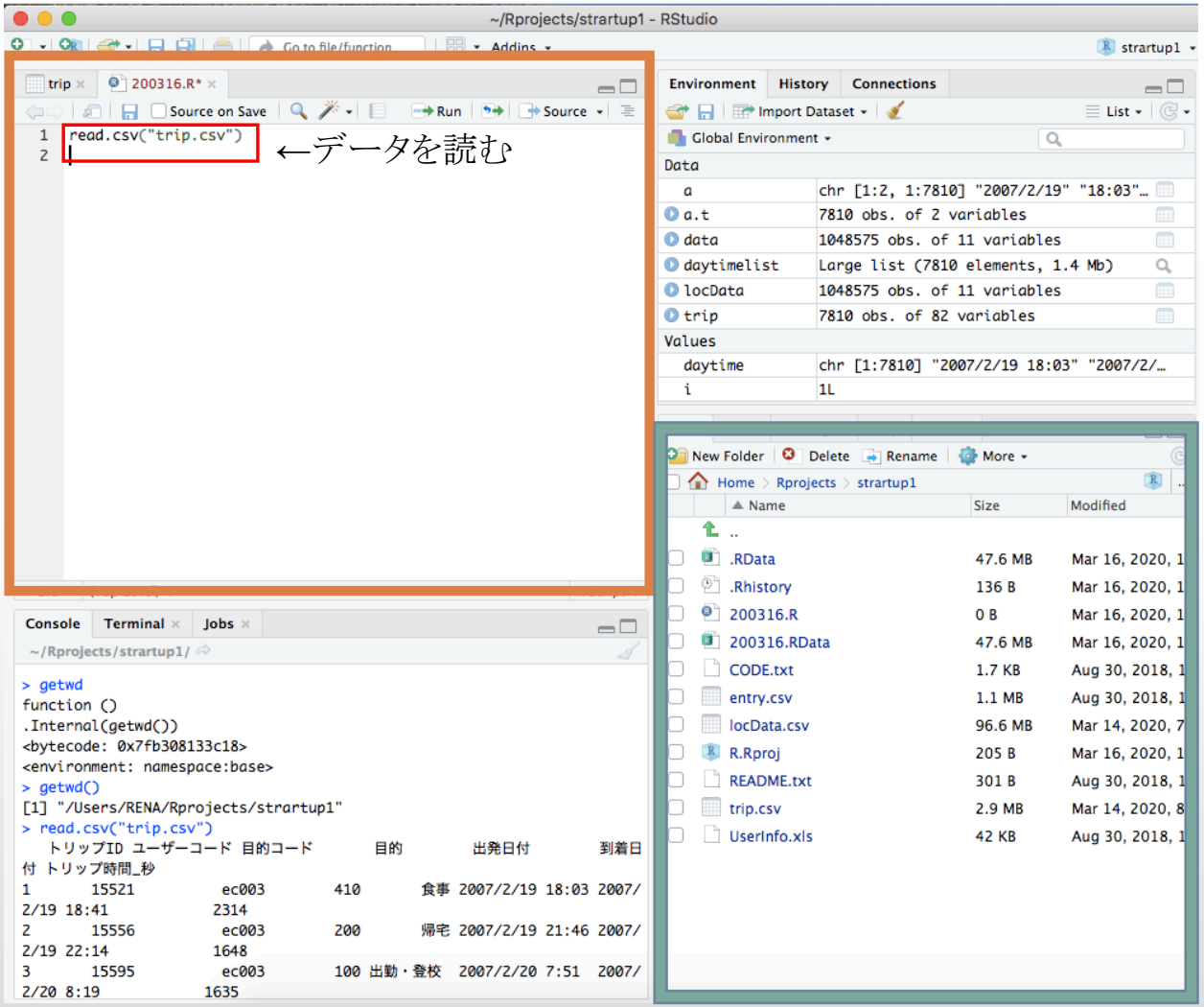
```
> getwd()
[1] "/Users/RENA/Rprojects/strartup1/start"
```

＊データのあるフォルダが作業ディレクトリと異なる場合，
変更しなければならない．作業ディレクトリと同じ場所にデータを入れておけば気にしなくても大丈夫．

- データの読み込み
  read.csv(“ファイル名”)

実際の操作

エディタ→
ここに書く



←fileやplotsなど.
☆必要なデータは作業しているprojectと同じディレクトリに入れておくと便利.
インプットしたいデータをクリックすると読み込める.

基本操作

□データフレームの要素へのアクセス

データの名前

```
> head(trip["目的"],n=5)  #「目的」列のはじめの5行だけ表示
          目的
1         食事
2         帰宅
3 出勤・登校
4           --
5 出勤・登校
```

□条件に合うデータの表示

```
trip$目的 #tripデータの目的列全部表示
trip[2] #tripデータの２列目を表示
trip[c(2,3)] #２列目と３列目を表示
trip[2,3] #2列3行目のデータ表示
trip[3,"目的"]  #「目的」列の3行目を表示
trip[trip$目的=="買い物",] #変数「目的」が「買い物」である行を表示
trip[trip$目的=="買い物" & trip$トリップ時間_秒>5000,]
#変数「目的」が「買い物」で，かつ，変数「トリップ時間__秒」が5000以上の行を表示

subset(trip,目的=="買い物",c(トリップID, 目的))
#変数「トリップID」と「目的」について，変数「目的」は”買い物”である行を表示
```

基本操作

□フィールドの追加

例えば，tripデータに「娯楽」列を追加したい．
娯楽:「目的」が，「買い物」「娯楽」「その他私用」の場合は1
それ以外の場合は0

```
cbind(trip, 娯楽 = ifelse(trip$目的=="買い物"|trip$目的=="娯楽"|trip$目的=="その他私用", 1, 0))
#これも同じ
transform(trip, 娯楽 = ifelse(trip$目的=="買い物"|trip$目的=="娯楽"|trip$目的=="その他私用", 1, 0))
```

＊他にもやり方はあるようなので，適宜調べてみてください．(withinなど)

□フィールドの削除

```
#トリップID,目的，移動手段の列を選ぶ
subset(trip, select = c(トリップID,目的,移動手段))

#作成フラグ，到着フラグの列を削除
subset(trip, select = -c(作成フラグ,到着フラグ))

#3列目から10列目を削除
head(subset(trip, select = -c(3:10)))
```

基本操作

- フィールドの値を更新
- 指定したフィールドに関してソート（orderを使う）
- フィールドの順番を変更
- 複数のデータフレームの合体

…

適宜調べてみましょう．すぐ出てきます．

ロケーションデータの条件抽出

移動目的が「買い物」かつ移動手段が「徒歩」のロケーションデータを抽出し，GIS上で可視化してみる

- もともとのロケーションデータには，**移動目的の列がない**
- しかも，loc dataは行数が多いので**Excelでは読みきれないからRでやるしかないーーー**

	トリップID	測位データID	移動手段	ユーザーコード	測位日時	緯度（日本測地系）	経度（日本測地系）	緯度（世界測地系）	経度（世界測地系）	測位モード	有効性
1	15375	2023091	バイク	ec029	2007/2/19 1:14	33.83487	132.7577	33.83817	132.7552	4	1
2	15375	2023092	バイク	ec029	2007/2/19 1:15	33.83045	132.7686	33.83375	132.7660	4	1
3	15375	2023093	バイク	ec029	2007/2/19 1:15	33.83506	132.7635	33.83836	132.7609	4	1
4	15375	2023094	バイク	ec029	2007/2/19 1:15	33.83484	132.7575	33.83814	132.7550	4	1
5	15375	2023095	バイク	ec029	2007/2/19 1:15	33.83060	132.7684	33.83390	132.7659	4	1
6	15375	2023096	バイク	ec029	2007/2/19 1:16	33.83487	132.7577	33.83817	132.7552	4	1
7	15375	2023097	バイク	ec029	2007/2/19 1:16	33.83682	132.7710	33.84012	132.7684	1	1
8	15375	2023098	バイク	ec029	2007/2/19 1:16	33.83692	132.7719	33.84022	132.7693	0	1
9	15375	2023099	バイク	ec029	2007/2/19 1:16	33.83685	132.7716	33.84015	132.7691	0	1
10	15375	2023100	バイク	ec029	2007/2/19 1:16	33.83701	132.7718	33.84032	132.7692	0	1
11	15375	2023101	バイク	ec029	2007/2/19 1:16	33.83711	132.7718	33.84041	132.7693	0	1
12	15375	2023102	バイク	ec029	2007/2/19 1:17	33.83698	132.7716	33.84028	132.7691	0	1
13	15375	2023103	バイク	ec029	2007/2/19 1:17	33.83705	132.7717	33.84035	132.7691	0	1

ロケーションデータ原本

ロケーションデータの条件抽出

```
#目的列が"買い物"で，かつ，移動手段が"徒歩"の行のみを抽出する．
SHP <- subset(cbtrip, cbtrip$目的=="買い物" & cbtrip$移動手段=="徒歩")

#上記で抽出した「徒歩で買い物」のトリップIDに対応するロケーションデータを抽出する．
(LocSHW <- merge(locData, SHP, by = "トリップID") )
```

	トリップID	測位データID	移動手段.x	ユーザーコード.x	測位日時	緯度（日本測地系）	経度（日本測地系）	緯度（世界測地系）	経度（世界測地系）	測位モード	有効性.x	ユーザーコード.y	目的コード	目的	出発日付	到着日付
1	15451	2035180	徒歩	ec011	2007/2/19 13:50	33.84329	132.7537	33.84659	132.7512	4	1	ec011	400	買い物	2007/2/19 13:25	2007/2/19 1
2	15451	2035191	徒歩	ec011	2007/2/19 13:53	33.84819	132.7513	33.85149	132.7488	4	1	ec011	400	買い物	2007/2/19 13:25	2007/2/19 1
3	15451	2035196	徒歩	ec011	2007/2/19 13:55	33.84822	132.7513	33.85152	132.7488	4	1	ec011	400	買い物	2007/2/19 13:25	2007/2/19 1
4	15451	2035195	徒歩	ec011	2007/2/19 13:54	33.84826	132.7513	33.85156	132.7488	4	1	ec011	400	買い物	2007/2/19 13:25	2007/2/19 1
5	15451	2035169	徒歩	ec011	2007/2/19 13:46	33.84432	132.7482	33.84762	132.7457	4	1	ec011	400	買い物	2007/2/19 13:25	2007/2/19 1
6	15451	2035182	徒歩	ec011	2007/2/19 13:50	33.84117	132.7555	33.84447	132.7530	1	1	ec011	400	買い物	2007/2/19 13:25	2007/2/19 1

□ CSVに書き出す

```
write.csv(LocSHW,"/Users/RENA/Rprojects/strartup1/start/LocSHW.csv")
```

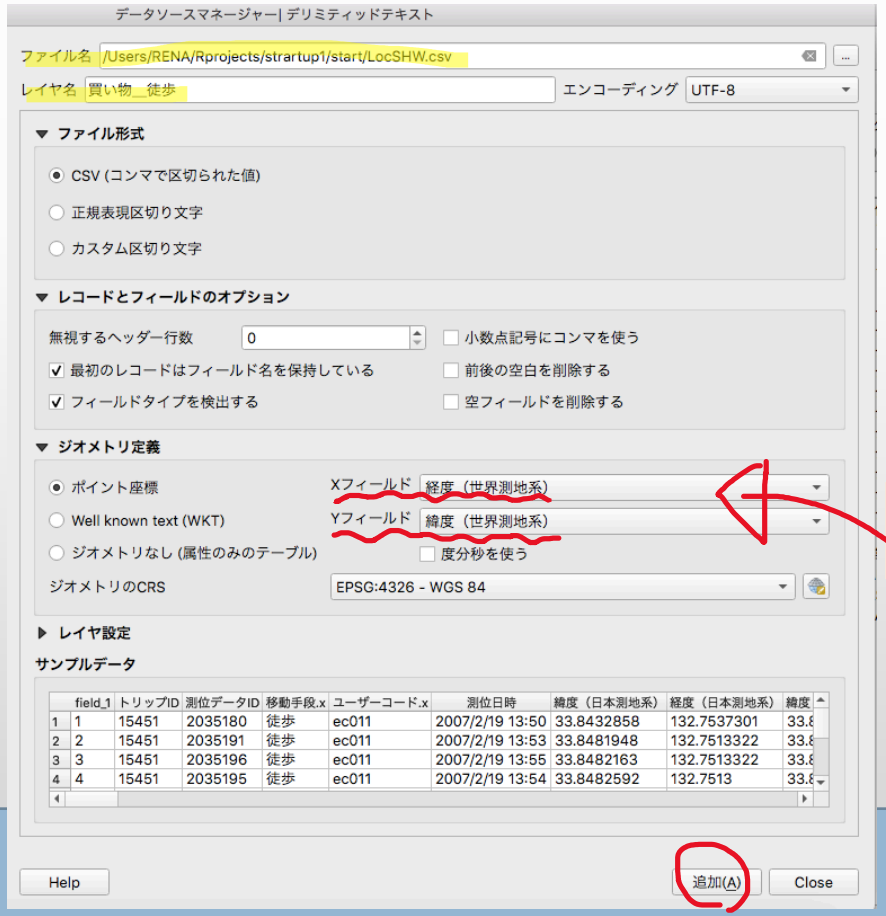
データフレーム名　　保存したい場所　　ファイルの名前を指定

出力したcsvファイルをQGISで可視化

「レイヤ」→「レイヤの追加」→「ディリミティッドテキストレイヤの追加」

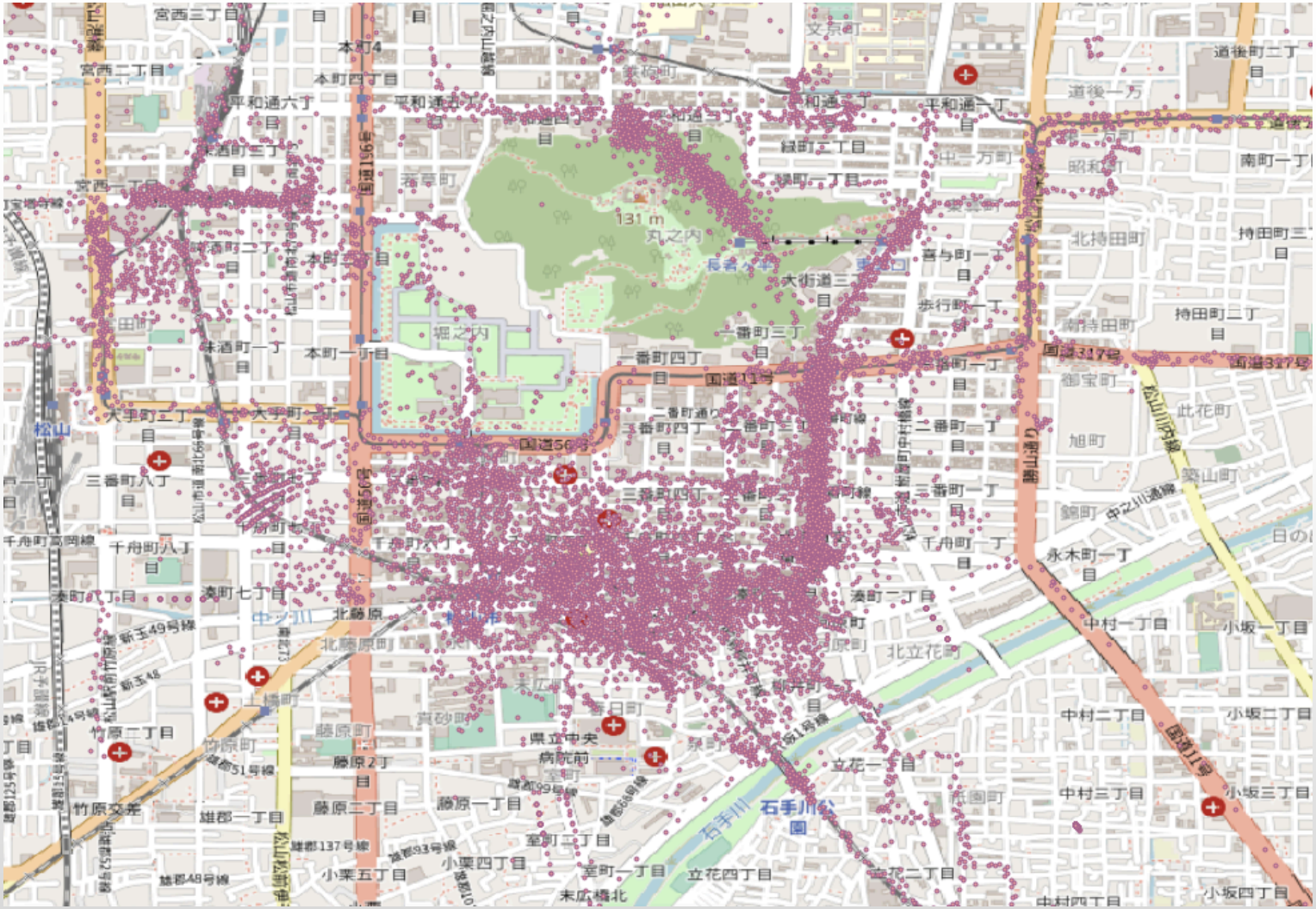
追加したいファイル（ここでは ocSHW.csv）のパスをファイル名に入力し，分かりやすいレイヤ名を付けます．

「ジオメトリ定義」の「Xフィールド」に出発地の経度の列を，「Yフィールド」に出発地の緯度の列を指定します．到着地を追加する場合も同様です．



Xフィールドに「経度（世界測地系）
Yフィールドに「緯度（世界測地系）

出力したcsvファイルをQGISで可視化



Rでピボットテーブル

前項で基本操作を含むデータ処理が便利にできる.
調べてみよう

```
install.packages("dplyr")
library(dplyr)
install.packages("tidyr")
library("tidyr")
```

- ✓ まず, データフレームの処理に便利なパッケージを読み込む
- ✓ %>%はdplyrパッケージで, 複数の処理を連結する演算子.
- ✓ %>%の左の値が, 右の値の第一引数として渡す.

```
> #目的と移動手段を軸にして，データの個数を集計
> trip %>%
+ dplyr::group_by(目的,移動手段) %>%
+ dplyr::summarise(データ数=n()) %>%
+ tidyr::spread(目的, データ数)
# A tibble: 12 x 10
```

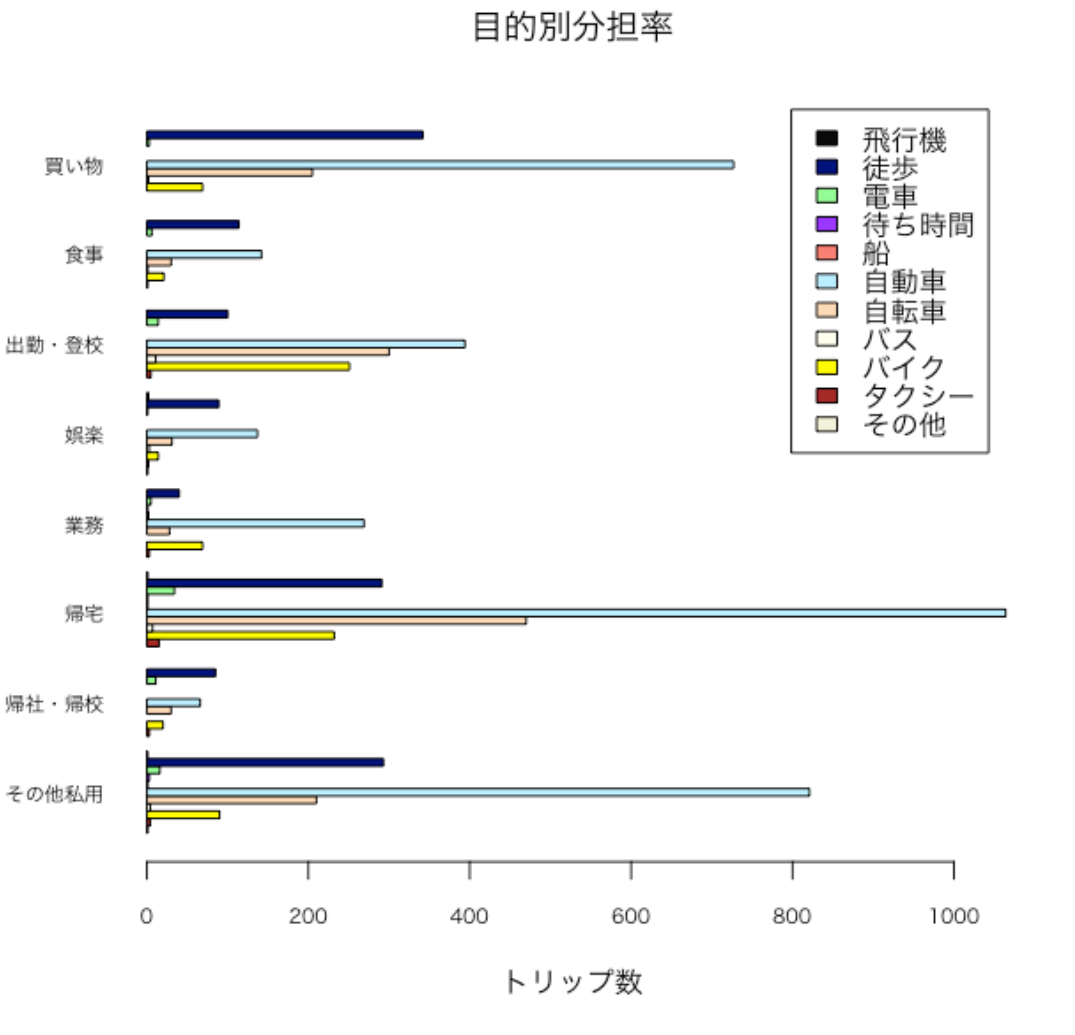
- 軸にしたい2つの変数でgroup_byする
- dplyrの中に入っているn( )という関数でデータ数をカウント
- spreadの中で,「列」にしたい軸変数名
(指定しなかった方が行になる)と, summariseの中で設けた集計変数名を指定

	移動手段	`---`	その他私用	`帰社・帰校`	帰宅	業務	娯楽	`出勤・登校`	食事	買い物
	<chr>	<int>	<int>	<int>	<int>	<int>	<int>	<int>	<int>	<int>
1	---	44	NA	NA	NA	NA	NA	NA	NA	NA
2	その他	NA	1	NA	NA	NA	1	NA	NA	NA
3	タクシー	4	4	3	15	3	2	5	1	NA
4	バイク	89	90	20	232	69	14	251	21	69
5	バス	2	4	NA	7	NA	3	11	1	2
6	自転車	96	210	30	470	28	31	300	30	205
7	自動車	307	821	66	1064	269	137	394	142	727
8	船	NA	1	NA	1	2	NA	NA	NA	NA
9	待ち時間	1	3	NA	1	1	NA	NA	NA	NA
10	電車	13	16	11	34	5	1	14	6	3
11	徒歩	44	293	85	291	40	89	100	114	342
12	飛行機	NA	1	NA	1	NA	2	NA	NA	NA

Rでピボットグラフ

```
install.packages("dplyr")  #データフレーム処理には読み込んでおくと便利
library(dplyr)
install.packages("tidyr")
library("tidyr")

#目的と移動手段を軸にして，データの個数を集計
pivot <- trip %>%
  dplyr::group_by(目的,移動手段) %>%
  dplyr::summarise(データ数=n()) %>%
  tidyr::spread(目的, データ数)

pivot <- pivot[-1,]  #1行目削除（行名--）
jiku <- pivot$移動手段 #1列目に行名が入っているので，これを保存しておく．（グラフの凡例に使うため）
                       #pivpt[,1]だと，データフレーム．ここではだめ．
                       #pivot$移動手段　だと変数．
pivot <- pivot[,-1:-2]  #1列目（行名）2列目削除（列名--）


#barchartで棒グラフを作る場合は，変数がベクトルか行列でなければならない
pivot <- as.matrix(pivot)
rownames(pivot) <- jiku  #行列の行名に先ほど保存した文字ベクトルのjikuを代入

#いよいよピボットグラフを描く．

par(las=1) #軸ラベルの向きを変える．0~3の4パターン
par(family= "HiraKakuProN-W3")  #グラフ内のテキストのフォント指定．日本語は文字化けするときは指定．

barplot(pivot, beside = T, legend = T, main="目的別分担率", xlab="トリップ数", horiz = T,
        cex.names = 0.7, cex.axis = 0.7, col= c("beige","brown","yellow1","ivory1","peachpuff",
                                                "lightblue1","salmon","purple1", "palegreen",
                                                "navyblue","grey6" ))
```

Rでピボットグラフ



	その他私用	帰社・帰校	帰宅	業務	娯楽	出勤・登校	食事	買い物
その他	1	NA	NA	NA	1	NA	NA	NA
タクシー	4	3	15	3	2	5	1	NA
バイク	90	20	232	69	14	251	21	69
バス	4	NA	7	NA	3	11	1	2
自転車	210	30	470	28	31	300	30	205
自動車	821	66	1064	269	137	394	142	727
船	1	NA	1	2	NA	NA	NA	NA
待ち時間	3	NA	1	1	NA	NA	NA	NA
電車	16	11	34	5	1	14	6	3
徒歩	293	85	291	40	89	100	114	342
飛行機	1	NA	1	NA	2	NA	NA	NA

本日の小課題

■ Rを用いて松山のPTデータ，PPデータの基礎集計を行い，データの傾向や気づきをまとめてください．

- Excelにも，ピボットテーブルのように簡単にデータの傾向を掴むことができるメリットがあります．RとExcelどちらも使ってみて，良い使い分けの感覚をつかみましょう．
- こういう処理がしたいということを，自分で手を動かしながら確認してみてください．

■ (最後に，基礎集計で得た気づきをGISを用いて可視化してください．)
*やりたい人だけで大丈夫です．GISに関する説明は，第1回フォルダ内の資料（ネットワークデータ作りのプロ須賀くん作成）や，
2018年度ゼミ資料　http://bin.t.u-tokyo.ac.jp/startup18/file/3-1.pdf
などを参考にしてください．

■ 以上の内容を，パワポ2-3枚程度にまとめて提出してください．

- 今回の1番の目的は，自分で色々調べながらデータ処理をしてみることで，Rに慣れることです．基礎集計を通して，何か大発見することが目的ではありません．
- ただし基礎集計が，次回扱うMNLで何に着目した推定をするかの，下ごしらえになるということは意識していただけるとありがたいです．