インターネットを活用した経済分析

第 2 日 作業の自動化，回帰分析，空間データ

山本 雅資　　伊藤 岳

University of Toyama
and
National Institutes for the Humanities

Email: gito@eco.u-toyama.ac.jp

February 16, 2017

Outline

R/プログラミングの用語と注意点

- (file) path: ファイルやディレクトリの「場所」, そこに至る経路
  - 例：伊藤の環境のデスクトップにある “sample” というディレクトリ (フォルダ) までの path は `/Users/Gaku/Desktop/sample`
  - 例：伊藤の環境のデスクトップにある “sample.csv” というファイルまでの path は `/Users/Gaku/Desktop/sample.csv`
  - Win での注意点：￥や “\” は “/” に置き換えること
  - path の取得方法：分からなければ,「自分の OS (Win 8 とか) ファイル パス 取得」等で Google!
- 拡張子：ファイル名の最後の “.” 以下の部分のこと
  - 例：“sample.csv” なら “.csv,” “sample.xls” なら “.xls”
  - ファイル名に拡張子が表示されていなければ, 「自分の OS (Win 8 とか) ファイル パス 取得」等で Google!
- 言語：日本語厳禁, 英語推奨
  - (R 言語や path 等を深く理解していない限り) R に読み込むファイル名やパスには日本語を絶対に含めてはならない
  - R を英語環境でインストールできるなら (してしまったなら) 英語環境を勧める (たぶんその方が質問対応もスムーズ)

R/プログラミングの用語と注意点

- R の実行：R に限らず，プログラミング言語を処理するシステムは，コードを「上から一行一行順番に実行していく」
  - したがって，途中の処理 A にエラーが出れば，処理 A を前提にしているそれ以後の処理にもすべてエラーが出る
  - 処理 A のエラーを修正して「処理 A の部分だけ」を回し直しても，コード全体は実行されない
  - 修正して「処理 A の部分」と「処理 A に依存する部分全部」を回し直さなければならない
- R のエラー・メッセージ：「何がおかしいか」「どんなエラーか」を教えてくれる
  - エラー・メッセージを Google すれば，簡単に解決策が出てくることも
  - 例：‘‘`Error: object 'x' not found`’’
  - このような「オブジェクトが見つからない」というエラーは，(1) 打ち間違いか (2) コードの実行順序を間違えた場合がほとんど
  - 打ち間違いは本人しか気付かない場合もあるので，よく見直すこと

研究の再現可能性

再現可能な研究

1. Reproduction: 同じデータ・方法を用いれば，誰が実行しても同じ研究結果が得られる
2. Replication: 同じ方法を異なるデータに適用しても，質的に同じ結果が得られる

「科学」性とプログラミング

- 経済学であれ政治学であれ社会学であれ，社会「科学」であるためには，reproducibility と replicability を満たさなくてはならない
- エクセルなど「ポチポチ」するようなソフトウェアの利用は，こうした条件を満たすことを妨げる
- 「ポチポチ」するようなソフトウェアの利用は，他人だけでなく，自分自身のためにもならない
  - 「一ヶ月前に行った解析」を「今」再現できる？

研究の再現可能性を担保するために

必要な記録

- データの出所，取得方法
- データの加工方法
- データの解析方法
- 解析結果の解釈
- これらすべてをコードで記述し，データセットとあわせて (原則) 公開する

研究成果の提出に際しては

- コードとデータセットを公開することが (原則) 必須
- 大学 (院) の課題でも，教員によってはコード提出を求める (ゲーム理論のテストで証明を書くのと同じ)

レポート (や卒論) を提出するまでに必要な作業

1. テーマ (e.g., 対象，問い，仮説，手法) を決める
2. データを取得する
3. データを解析しやすいよう整理・加工する
4. データを解析する (e.g., 回帰分析)
5. 解析結果を解釈する
   - 回帰分析の結果は表だけで解釈するのは無理があるので，作図する
   - 「何かおかしいところがないか」「改善点はないか」を確認して，あれば修正する
6. 文章を書き，推敲 (修正) する

コードを書かないと困る理由

1. 全部「ポチポチ」する必要がある
   - ファイルを開いて，変数をクリックして，ドラッグして，etc.
2. 「先週やったこと」を再現し，修正するのは至難の技
   - 詳細な細かなメモでもない限り，先週やったことは分からない．最終的な解析結果や図はあっても,「どうやったか」なんて覚えてない
   - たとえば，指導教員から「この図見にくいから綺麗にして」とか「この回帰分析は，ある変数を投入した場合，しない場合の結果を提示しなさい」とか突然言われても対応できない (こういう要求をギリギリに出す教員もいる [かも知れない])
3. 結果を見て「何かおかしい」と思っても,「どこで間違ったか」が分からない (覚えてない)
   - だからといって虚偽 (捏造) の報告をすれば，学生なら停学もしくは退学，研究者なら失職
4. 適切な図の作成や解析はできないし，図や表も汚く見れたものではない (教員によっては減点要素)

「ちゃんと (R で) コードを書いてる人」だったら大丈夫

1. 一連の作業はコードになっているので，1 週間後でも 1 年後でも，原則確認・再現できる
2. 「先週やったこと」の再現・修正もすぐにできる (「後で見て分かる」コードがあれば)
   - たとえば，指導教員から「この図見にくいから綺麗にして」とか「この回帰分析は，ある変数を投入した場合，しない場合の結果を提示しなさい」とか突然言われても，小一時間あれば大抵対応できる
   - なんなら，寝る前にコードを回しておけば，朝起きたらできてる
3. 結果を見て「何かおかしい」と思えば，コードの処理を追って確認すればいい
   - 翌日でも，1ヶ月後でも，1 年後でも問題ない (ただし,「後で見て分かる」コードなら)
4. R なら，最新の手法もすぐにパッケージで利用可能．また，図や表も専用のパッケージを用いて美しく出力できる

R による作業の記録と自動化

R によって，いずれの作業も自動化できる

- データの取得
- データの加工
- データの解析
- 解析結果の表・図の作成

たとえば，

- データの取得と加工：毎日更新されるデータを自動で取得し，一定のフォーマットに加工する
- データの解析：あるデータセットについて数百以上の回帰分析を回して，結果を検討する
- 解析結果の表・図の作成：全ての説明変数について，限界効果 (marginal effect ← 今日説明) の図を作成して保存する

R による作業の自動化

関数

- 特定の (一部の) 作業を自動化する
- 作業の例：「ある統計量を計算する」「与えられたデータについて，回帰分析を実行する」
- コードの例：次スライド

R スクリプト

- 一連の作業全体を自動化する
- 作業の例：「特定のデータを取得して，解析をして，プロットする」
- 処理の一括実行には必須
- 後述の R Markdown と組み合わせることで，作業の記録にもなる

R による作業の自動化：関数

(不偏) 分散を求める

- ある変数 x の不偏分散 (unbiased variance) は $u_x^2 = \frac{\sum_{i=1}^{n}(x-\bar{x})^2}{n-1}$
- ある変数 (母集団 or 標本の) x の分散は $\sigma_x^2 = \frac{\sum_{i=1}^{N}(x-\mu)^2}{N}$

R による実行

- 正規分布に従う乱数を発生させ，`x` というオブジェクトに格納
- デフォルトの `var()` 関数を使って，不偏分散を求める
- (1) R のエディタ (or R 対応のプログラミング用エディタ) にコードを記述．(2) 実行したい部分を選択し，`command + enter` (win では `ctrl + R`) で実行可能 (コンソールの出力は以下の通り)

```
> x = rnorm(100, mean = 0, sd = 1) ## normal distribution
> head(x) ## skim the first 6 entries
[1] 0.42721475 -0.81931186 1.14792710 2.21890227 -0.35855223 0.08595286
> var(x) ## compute unbiased variance
[1] 1.068102
```

R による作業の自動化：関数

デフォルトにはない作業を行なう

- var() 関数は不偏分散 $u_x^2 = \frac{\sum_{i=1}^{n}(x-\bar{x})^2}{n-1}$ を求める関数なので，母集団の分散 $\sigma_x^2 = \frac{\sum_{i=1}^{N}(x-\mu)^2}{N}$ は自分で計算しなくてはならない
  - 母集団の平均 μ は標本の平均 $\bar{x}$ と同じく mean() 関数で求められる

R での実行：逐一計算する

```
> sum((x - mean(x))^2) / length(x)
[1] 1.057421
```

R での実行：新しい関数を定義する

```
> variance <- function(argument) {
    output <- sum((argument - mean(argument))^2) / length(argument)
    return(output)
}
> variance(x)
[1] 1.057421
```

R による作業の自動化：関数

関数の構成要素

- 引数：argument．ベクトルでもデータフレームでも何でもよい
- 処理：output を計算している 2 行目．引数と整合的なものでなければならない
- 返り値：output．最終的に得られる出力結果．return() 関数に与える引数で指定する

関数定義のありがたみ

- 新たな関数 variance() を定義することで，コードを簡略化可能
- 「ある引数について一定の処理をする」作業は，基本的に関数化できる
  - 「よくやる作業」は自動化してしまえる
  - ファイルの読み込み，加工，解析，作図，etc
  - より高度／便利な関数：後ほど

R による作業の自動化：R スクリプト

R スクリプト

- 「ある (完結した) R のコード全体」のこと
- 例：前出の R コードを，variance.R として保存する (“>” は不要)

```
1 variance <- function(argument) {
2     output <- sum((argument - mean(argument))^2) / length(argument)
3     return(output)
4 }
5 x <- rnorm(100, mean = 0, sd = 1) ## normal distribution
6 variance(x)
```

- source() 関数で実行すれば，variance.R の処理を実行できる

```
1 > source("variance.R")
2 [1] 1.057421
```

- 関数定義，正規乱数の生成，母集団の分散を求める，という作業を，source(‘‘variance.R’’) と打つだけで自動実行できた！
- データ読込み・加工・解析・作図のような複雑な作業も自動化可能

R による作業の自動化：R スクリプト

R による実行

- R スクリプトは，よく使う関数を呼び出す，簡略な「俺俺パッケージ」にもなる
- 例：variance() 関数を variance_function.R として保存する

```
1 variance <- function(argument) {
2     output <- sum((argument - mean(argument))^2) / length(argument)
3     return(output)
4 }
```

- source() 関数で variance_function.R を読み込めば，variance() 関数をいろんな場面で使える

```
1 > x <- rnorm(100, mean = 0, sd = 1) ## normal distribution
2 > variance(x)
3 Error: could not find function "variance"
4 > source("variance_function.R") ## load variance() function
5 > variance(x)
6 [1] 1.057421
```

R による作業の記録：R Markdown を用いた文芸的プログラミング

文芸的プログラミング (literate programming)

- R スクリプトは便利だが，文章による説明が不足する (後で読んで分かりにくい)
- コメントアウト (R 言語では# の後の箇所) した部分に説明を加えられるが，R による実行結果を同時にみることができない
  - どういう処理がされているか，処理が回るか等が確認しにくい
  - 人と共有するときにも，説明が大変
- 文芸的プログラミング：コードと，自然言語によるコードの説明・解釈を並行して書く技法
  - (1) コード，(2) 自然言語による説明，(3) コードの実行結果を 1 つのファイル (e.g., html, pdf) にまとめて記録できる！
- R では，`R Markdown` と呼ばれる言語体系で実装されている
  - 講義資料の html ファイルは，すべて R Markdown によって作成

R による作業の記録：R Markdown を用いた文芸的プログラミング

R Studio と R Markdown

- R Markdown は，R Studio (R の兄弟アプリ) で実装されている
- `https://www.rstudio.com` に行って，インストールする
- R Studio は，R の実行環境 $+ \alpha$ の機能を提供する
- 当然，R Markdown だけでなく，通常の解析にも使える (R か R Studio かは好み)

R Markdown 事始め

1. File $\rightarrow$ New File $\rightarrow$ R Markdown を選択
2. 新規作成ダイアログが開くので，Document を選択
   - ファイル名や著者，出力フォーマット等を設定
   - “html” を選択 (“pdf” にしていいのは LaTeX を使いこなせる人だけ)
3. 保存して，`command + shift + k`
4. サンプルファイルが生成される

R Markdown = 統計処理言語 R + マークアップ言語 Markdown

Markdown 記法による文章の記述

- Markdown: html と同様の「マークアップ言語」の一種
- 手軽に文章構造を明示でき，対応アプリも多数
- 専用のスクリプトを使えば，html のような他のマークアップ言語に容易に変換可能
  - Markdown 記法の日本語での解説：日本語 Markdown ユーザー会 (`http://www.markdown.jp`)
  - html, LaTeX, MS Word 等への変換：Pandoc (`http://pandoc.org`)

R コードチャンク

- R コードの部分のこと．```` ```{r hogehoge, options} R code``` ```` と書く
- 各チャンクには，固有の名前 (hogehoge) をつける (べき)
- `R code` の部分には，通常の R コードをそのまま書けばよい

課題：R Markdown を用いた文芸的プログラミング

R Markdown を触ってみる (習うより慣れろ)

1. R コードの部分と自然言語による記述 (入力，左側) を，html の出力 (右側 or ブラウザ) と対応させながら「解読」する
2. 分かったら，自然言語の文章を変更・追記して，`command + shift + k` してみる (Win では `ctrl + shift + k`)
3. 次いで，次スライドの R コードを追記して，`command + shift + k` してみる
   - どこに足せば良いかも考える
   - 色々文章や R コードを足して遊んでみる
   - `command + shift + k` ではなく `command + shift + s` とするとどうなるか試してみる

R Markdown の日本語での解説

1. `http://gihyo.jp/admin/serial/01/r-markdown`
2. `http://kohske.github.io/R/rmarkdown/`

課題：R Markdown を用いた文芸的プログラミング

パッケージの呼び出し

````
```{r library, message=FALSE, warning=FALSE}
# install.packages(stargazer, dependencies=TRUE)
# install.packages(tidyverse, dependencies=TRUE)
library(stargazer)
library(tidyverse)
```
````

サンプルデータ iris の呼び出しと線形回帰

````
```{r regression}
data(iris)
summary(iris)
fm = as.formula(Sepal.Length ~ Sepal.Width + Petal.Length + Petal.Width +
      Species)
reg = lm(fm, data=iris)
stargazer(reg, type="text", single.row=TRUE)
```
````

課題：R Markdown を用いた文芸的プログラミング

R Markdown ファイルから R コードを抽出する

```
> # install.packages(knitr, dependencies=TRUE)
> library(knitr)
> path2rmarkdown = "path2yourRmarkdown.Rmd"
> purl(path2rmarkdown) ## ワーキングディレクトリにファイルが保存される
```

提出物：課題 **2–1**

- R Markdown のソースファイル (.Rmd ファイル) および出力 (.html ファイル)
- 前スライドのコードに加えて，`command + shift + k` の結果，`command + shift + s` の結果がどう違うかも書く
- メール添付で提出

データ取得：ウェブスクレイピングと API

データの取得

- 多くの研究では，政府機関や研究機関が作成・公開しているデータセットを入手・結合・加工して，解析を行なう
- ウェブから入手できるデータの場合，ブラウザ上で「ポチポチして」データを取得することもできるが...
- 大変！ たとえば，株価や為替のように逐次変動するデータを用いる場合，変動する度に (e.g., 毎時間・毎日！) 「ポチポチ」してダウンロードし，保存・解凍する必要がある．さらに，文字の情報しかないかもしれない (何百何千何万回もコピペする？)
- 「ポチポチ」は手間もかかれば，ミスも出るし，時間の無駄！

コードによる自動化

- R でコードを書けば，「ポチポチ」する作業は完全に省略できる
- 手間・時間が省け，かつミスをなくせる
- 方法としてのウェブスクレイピングと API

データ取得：ファイルの読み込み (一部復習)

R におけるファイルの読み込み

1. ファイルパスの取得：ファイルパス (file path; 「場所」) を指定する
   - パスの取得：(1) Mac OS 10.12 は右クリック + `option`, (2) Win 8 は `shift` + 右クリック
2. パスを関数に渡す：取得したパスでファイルを指定し，拡張子にあった関数を用いて読み込む

```
> library(readr) ## load package
> path <- "/Users/Gaku/Dropbox/sample.csv" ## file path
> data <- read_csv(path) ## read
```

その他の操作

- サンプルデータの呼び出し
- データの書き出し (保存)
- ウェブ上のファイルの読み込み
- R コード: `1_read_and_write_files.html`

データ加工：オブジェクトと要素へのアクセス (一部復習)

R におけるオブジェクト

- 主要クラス：ベクトル (vector)，行列 (matrix)，データフレーム (data.frame)，リスト (list)，配列 (array)
- この講義では，配列を除く上記のクラスと，データフレームの類似クラスを扱う (他にも色々ある)

オブジェクトの各要素へのアクセス

- たとえば，(4，5，2，3，1) という数列を格納したベクトル型 (クラス) オブジェクトを定義し，その 2 番目の要素を取り出すには，次のコードをコンソールに入力する

```
> vec = c(4, 5, 2, 3, 1)
> vec[2]
[1] 5
```

- 行列やデータフレーム，リストの要素へのアクセス？
- R コード: `2_objects.html`

データ取得：ウェブスクレイピング

ウェブスクレイピングとは

- ウェブサイト上の情報を抽出する技術や技法
- ウェブサイト上の文字情報やハイパーリンク等を「削り取る・かき集める (scrape)」ことで取得する

R による実行

- 「リンク先のファイル」の自動取得：大量の zip ファイルを取得する
- 文字情報の自動取得：為替データをリアルタイムに取得する
- R コード: `3_scraping.html`

データ取得：API (Application Programming Interface)

API とは

- Application Program Interface
- ソフトウェア間で機能を共有する仕組み (命令，関数等の集合)
- 遠隔地にあるコンピュータやサーバーが提供する機能・データを，他のソフトウェア (ここでは R) で利用できる
  - API 自体については，「API とは 解説」等で Google 先生に聞く (さぼってる訳ではなく，授業時間が足りないから)

R による実行

- UN Comtrade のデータを API を用いて取得する
- 昨日は「ポチポチ」してデータを取得したが，これを自動化する (R の関数と Comtrade の API を使って取得する)
- R コード: `4_api.html`

回帰分析 (復習) とシミュレーション

現実のデータを用いた解析

- カナダのセンサスデータ (1971) `Prestige`
- `car` パッケージから読み込めるサンプルデータ

シミュレーション

- 疑問：モデルを正しく特定できている (かつ種々の仮定も満たされている) として，回帰分析は「真の」影響の大きさとその不確実性を，本当に教えてくれるのだろうか
- 検討：「真のモデル」が分かる仮想データを大量につくり，解析
- 手順：「真のモデル (データ生成プロセス data generating process) は同じだが，微妙に異なる」$n = 1,000$ の仮想データを $3,000$ 個つくり，$3,000$ 回の回帰分析を回し，その結果を俯瞰する
- 問題：「ポチポチ」してたら日が暮れる (どころじゃ済まない．っていうか無理ゲー．3,000 どころか 3,000,000 回かもしれない)

回帰分析 (復習) とシミュレーション

線形回帰モデルの数式表現

- 線形回帰モデルは，次のように表現できる

$$\begin{aligned} y_i &= \boldsymbol{X}_i\boldsymbol{\beta} + \epsilon_i \\ &= \beta_1 X_{i1} + \cdots + \beta_k X_{ik} + \epsilon_i, \quad \text{for } i = 1, \ldots, n, \end{aligned} \tag{1}$$

where $\epsilon_i \sim N(0, \sigma^2)$ denotes an *IID* (independent and identically distributed) error term, and $X_{i1} = 1$ is a constant term

- また，次のようにも表現できる．今から検討するシミュレーションは，この表現を念頭に置くと理解しやすい

$$y_i \sim N(\boldsymbol{X}_i\boldsymbol{\beta}, \sigma^2), \quad \text{for } i = 1, \ldots, n. \tag{2}$$

- あるいは，

$$\boldsymbol{y} \sim N(\boldsymbol{X}\boldsymbol{\beta}, \sigma^2\boldsymbol{I}), \tag{3}$$

where $\boldsymbol{I}$ is an $n \times n$ identity matrix (単位行列).

回帰分析 (復習) とシミュレーション

R による実行

- サンプルデータ `Prestige` の回帰分析
- シミュレーションによる回帰分析の理解
  - 3,000 回だろうが 3,000,000 回だろうが一箇所変えるだけ
- R コード: `5_regression.html`

提出物：課題 2–2

- `5_regression.html` の「2.4.3 課題」で指示したシミュレーション (母数の推定) を行なう
- 2 つのシミュレーションから得られたヒストグラムと，それぞれのシミュレーション結果の解釈を付す
- 古いマシンを使っている場合は，シミュレーションの回数を減らしても可

回帰係数と限界効果

線形回帰における (偏) 回帰係数の解釈

$$\hat{y} = \beta_0 + \beta_1 x_1 + \beta_2 x_2 + \beta_3 x_3 + \epsilon \tag{4}$$

という回帰式を考える

- この回帰式において，独立変数 x_k の偏回帰係数 β_k は，他の独立変数の値を一定に保ったとき，x_k が 1 (単位) 増加したときの従属変数 (の予測値) $\hat{y}$ の増加量
- 重回帰分析における「偏回帰係数」の「偏」は偏微分の「偏 (partial)」と同じ意味
- つまり,「偏」は「他の (独立) 変数の値を一定に保ったとき」の意味
- 「x_k が 1 (単位) 増加したときの従属変数 (の予測値) $\hat{y}$ の増加量 (影響の大きさ)」のことを，限界効果 (marginal effect) という
  - 回帰係数と同じに思えるが，より高度な一般化線形モデル (本講義では扱わない) の理解や，後述の交互作用項を含むモデルの理解には限界効果が非常に重要

回帰係数と限界効果

線形回帰における (偏) 回帰係数の解釈

- 解析では，統計的有意性 (statistical significance) と同程度以上に，実質的有意性 (substantial significance; 効果の大小) が重要
- 実質的有意性を解釈する上では，限界効果が鍵になる
  - 一般化線形モデル (identity link 以外) では線形回帰のような直感的な解釈が困難なため，限界効果の計算・作図がより重要になる

限界効果

- 「偏」の字から直感的にわかる通り，x_k の限界効果は (4) 式を x_k で偏微分すれば求められる．すなわち，

$$\frac{\partial \hat{y}}{\partial x_k} = \beta_k \tag{5}$$

回帰係数と限界効果

限界効果，実質的有意性の重要さ

- よく聞く表現：「この変数はトウケイテキニユーイだった．大事だ」
- 気をつけるべき点：統計的に有意だからといって，実質的に有意とは限らない (「統計的に有意でも，大して影響のない」変数もある)

「他の独立変数が一定のとき」の意味をよく考える

- 常に「一定に保つ」ことができる訳ではない
  - 例：二乗項や交互作用項／交差項 (interaction term) が含まれる回帰式 (次スライド)
  - 交互作用 (interaction)：ある独立変数の効果が，他の独立変数の値によって異なること
- いずれの場合でも，ある独立変数 x_k「だけ」を動かして，その限界効果を提示することは不可能 (掛け算してるから！)
- さらに，観察データ (observational data) では，往々にして独立変数間に相関がある (Hanmer & Ozan Kalkan, 2013)

限界効果の計算：Interaction term がある場合

交互作用項とは

- $x_1 \times x_2$ のような,「かけ算」で表現される (独立) 変数
- 交互作用項 $x_1 \times x_2$ を “product term,” x_1 と x_2 を “constitutive term(s)” (main effect) と呼ぶ
  - 理論的議論：Berry et al. (2010, 2012); Brambor et al. (2006); Clark et al. (2006); Kam & Franzese (2007); Rainey (2015)

交互作用項の解釈と限界効果

- よくある間違い (1)：x_1 (e.g., 投薬の有無) や x_2 (e.g., 性別) の交互作用項 $x_1 \times x_2$ を回帰式に投入したから，x_1 と x_2 をバラバラに回帰式に投入する必要はない！
  - 誤：$y = \beta_0 + \beta_3 x_1 x_2 + \epsilon$
  - 正：$y = \beta_0 + \beta_1 x_1 + \beta_2 x_2 + \beta_3 x_1 x_2 + \epsilon$
- よくある間違い (2)：交互作用項の回帰係数 β_3 が有意だった！ だから，x_1 や x_2 単独ではなく，その組み合わせが重要なんだ！ (もうちょっと作業が必要)

限界効果の計算：Interaction term がある場合

- 交互作用項がある場合，product term も constitutive term も，単一の回帰係数のみでは解釈できない
- たとえば，$y = \beta_0 + \beta_1 x_1 + \beta_2 x_2 + \beta_3 x_1 x_2 + \epsilon$ の回帰式を考える
- x_1 の限界効果を計算するには回帰式を x_1 で偏微分すればよいので，

$$\frac{\partial y}{\partial x_1} = \beta_1 + \beta_3 x_2 \tag{6}$$

- すなわち，x_1 の限界効果は β_1 ではなく，$\beta_1 + \beta_3 x_2$
- x_2 の限界効果も同様に求められる

$$\frac{\partial y}{\partial x_2} = \beta_2 + \beta_3 x_1 \tag{7}$$

- x_2 の限界効果は β_2 ではなく，$\beta_2 + \beta_3 x_1$
- $y = \beta_0 + \beta_1 x_1 + \beta_2 x_1^2 + \epsilon$ のような，二乗項 (“2” である必要はない．3 以上でもよい) を含む回帰式についても同様

限界効果の計算：Interaction term がある場合

含意

(1) x_1 と x_2 の「掛け算」が入るのだから，x_1 を「だけ」を変化させ，「他の変数を一定に」することはできない

(2) $x_2 = 0$ (かつ／あるいは $\beta_3 = 0$) でない限り，x_1 の限界効果は x_2 の水準に依存する

(2') x_2 の水準を固定しなければ x_1 の限界効果を計算できないが，x_2 の水準によって x_1 の限界効果が変化する

- 実際問題，「様々な x_2 の水準における x_1 の限界効果を示す図」がなければ解釈できない！
- 典型的には，x_2 を x 軸に，x_2 がある値のときの x_1 の限界効果を y 軸にとったグラフを使う

(3) 一般化線形モデル (identity link 以外) の場合，“compression effect” が生じるため，さらに大変 (「他の全ての独立変数」の値に依存する; Berry et al., 2010; Rainey, 2015)

限界効果の計算と作図：自動化

R による実行

- ある独立変数について限界効果を計算し，図を描く
- 交互作用を含むモデルについても，同じことをする
- `interplot`，`sjPlot` パッケージを用いて作図する
- R コード: `6_effect.html`

空間データ：新たな情報と課題

空間データとその「ありがたみ」

- ここまで扱ってきたデータは,「国」のような情報を持っていても, それが空間的・地理的にどのように配置されているか, といった地理空間情報をもっていない
- これに対して, 地理的・空間的な「場所」に関する情報を保持するデータを, 一般的に「空間データ (spatial data)」と呼ぶ
- 空間データは, 非空間データがもたない「データの場所」や「データ間の距離・近接性」といった新たな情報を教えてくれる
- 空間データを用いることで, たとえば「ホットスポット」の解析や「流行の地理的拡散」のような地理空間と切り離せない現象を解析できる
- 地理空間を無視した従来の解析が見落としていた, 重要な要因を見つけれるかもしれない

空間データ：新たな情報と課題

「ありがたみ」の裏返し

- とはいえ，新たな情報が利用できるということは，分析上注意を要する点が増えるということでもある
- 特に，回帰分析を行なうときには，データの地理空間情報が推定に与え得る影響に注意する必要がある
  - 本講義では踏み込まないが，興味があれば「地理学の第一法則」「空間統計モデル」「空間計量経済モデル」で Google してみる

解析上の問題の例

- 残差の空間的偏り：回帰分析の残差が，空間的に偏ることがある (e.g., ある地域には負の値が，ある地域には正の値が固まる)
- 空間的自己相関 (spatial autocorrelation)：「距離が近い程，物事の性質が似る／異なる」傾向
- 非空間データを想定した回帰モデルでは適切に推定できない可能性がある (誤差の独立性 [の違反]，不均一分散や欠落変数バイアス)

空間データ：新たな情報と課題

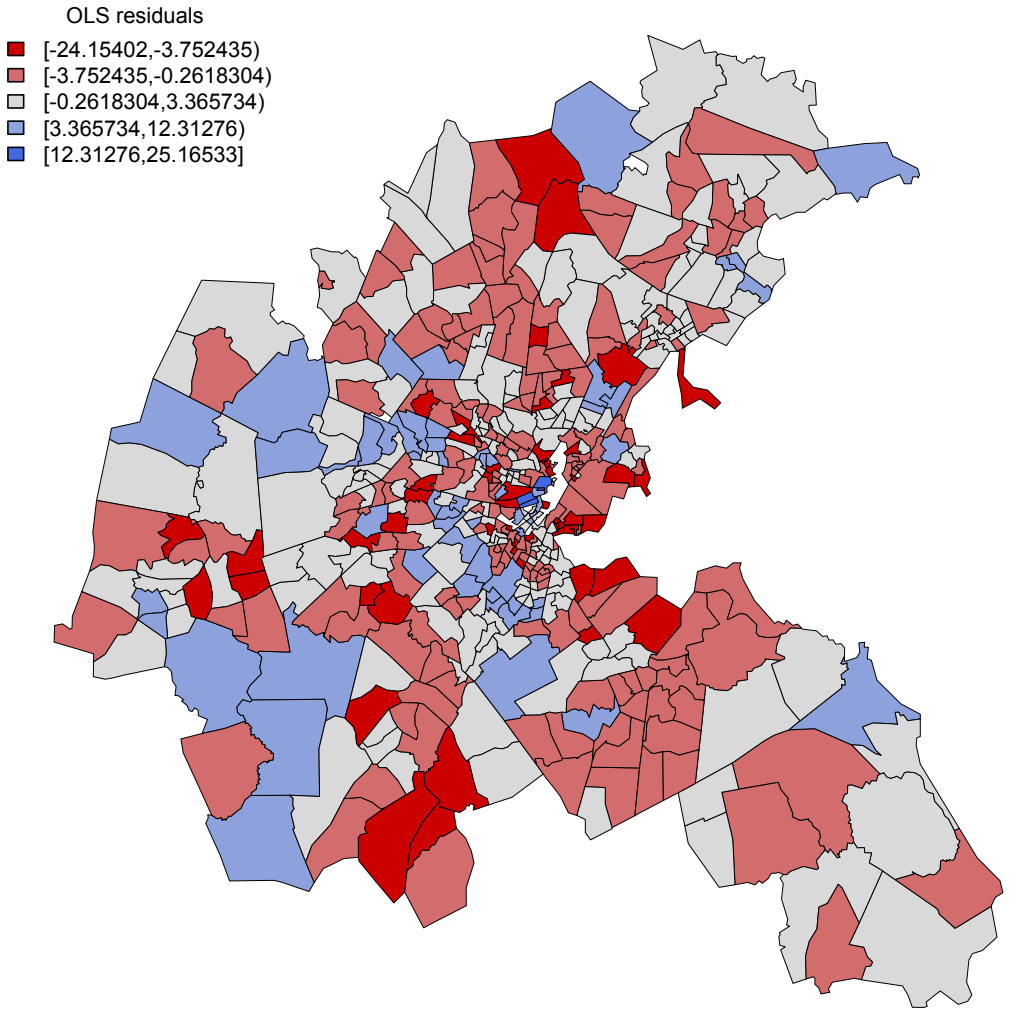
実際的な問題

- 地理空間情報や GIS (geographic information system) 処理用ソフトウェアは統計ソフトウェア以上に高価で，個人では購入困難 (e.g., ArcGIS)
- フリーのソフトウェアも発展してきてはいる (e.g., QGIS)
- しかし，空間データの統計処理手法は発展途上で，新たな手法が次々に提案されているため，ソフトウェアが追いついていない
- 通常の統計処理以上に，繰り返し処理や時間・計算負荷のかかる処理 (= 寝てる間にコンピュータにやらせておきたい処理) が多い

R の活用

- R なら新たな手法もパッケージですぐに利用でき,「寝ている間の処理」もできる
- 「R でも空間データの処理ができる」というより,「空間データの処理にこそ R を使うべき」

空間データ：残差の空間的偏り



空間データ：残差の空間的偏り

線形回帰の結果，残差，空間

- 線形回帰の残差が空間的に偏っている (ように見える)
- 疑問：(1) 本当に (統計的に有意に) 偏っているのか，また (2) データの空間構造に配慮した回帰モデルは，R で実行できるのか
- R コード: `7_spatial.html`

Berry, William D; Jacqueline H R DeMeritt & Justin Esarey (2010) Testing for Interaction Effects in Binary Logit and Probit Models: Is an Interaction Term Essential? *American Journal of Political Science* 54(1): 248–266.

Berry, William D; Matt Golder & Daniel Milton (2012) Improving Tests of Theories Positing Interaction. *Journal of Politics* 74(3): 653–671.

Brambor, Thomas; William Clark & Matt Golder (2006) Understanding interaction models: Improving empirical analyses. *Political Analysis* 14(1): 63–82.

Clark, William R; Michael J Gilligan & Matt Golder (2006) A simple multivariate test for asymmetric hypotheses. *Political Analysis* 14(3): 311–331.

Hanmer, Michael J & Kerem Ozan Kalkan (2013) Behind the Curve: Clarifying the Best Approach to Calculating Predicted Probabilities and Marginal Effects from Limited Dependent Variable Models. *American Journal of Political Science* 57(1): 263–277.

Kam, Cindy D & Robert J Franzese (2007) *Modeling and Interpreting Interactive Hypotheses in Regression Analysis: A Refresher and Some Practical Advice*. Ann Arbor, MI: University of Michigan Press.

Rainey, Carlisle (2015) Compression and Conditional Effects: A Product Term Is Essential When Using Logistic Regression to Test for Interaction. *Political Science Research and Methods forth*.