

貿易データの分析

田中 鮎夢

2022-12-05

1 貿易データの読み込み

trade2013.csv (<https://ayumu-tanaka.github.io/teaching/trade2013.csv>)には、2013年の世界の2国間の貿易データが含まれている。含まれている変数は以下の通りである。

- 輸出国: country1
- 輸入国: country2
- 輸出額の対数値: ltrade
- 輸出国のGDPの対数値: lgdp1
- 輸入国のGDPの対数値: lgdp2
- 輸出国と輸入国の間の距離: ldist

```
library(readr)
trade2013 <- read_csv("trade2013.csv")
head(trade2013)
```

```
## # A tibble: 6 × 6
##   country1      country2      ltrade lgdp1 lgdp2 ldist
##   <chr>        <chr>      <dbl> <dbl> <dbl> <dbl>
## 1 United States United Kingdom  10.5   30.3  28.4  8.37
## 2 United States Austria         8.54   30.3  26.5  8.54
## 3 United States Belgium-Luxembourg 10.0   30.3  26.8  8.45
## 4 United States Denmark         8.20   30.3  26.0  8.45
## 5 United States France         10.3   30.3  28.4  8.47
## 6 United States Germany        11.0   30.3  28.7  8.48
```

なお、<- と = はほとんど同じである。Mac では「option」 + 「=」 で入力すると、便利である。（断り: 本来、= に統一した方が手間が省けるが、本文書では従来からの <- を惰性で使っていることが多い。特に意味はない。）

- 参考: What are the differences between “=” and “<-” assignment operators? (<https://stackoverflow.com/questions/1741820/what-are-the-differences-between-and-assignment-operators>)。

2 回帰分析 (lm)

「trade2013」というデータを使って、ltradeを従属変数、lgdp1、lgdp2、ldistを説明変数とする式について、最小2乗法実行し、「model1」という名前で記憶して下さい、というコマンドを実行する。

```
model1 <- lm(ltrade ~ lgdp1 + lgdp2 + ldist, data = trade2013)
```

3 推定結果の表示 (summary)

保存された分析結果「model1」を表示するコマンドを実行する。

```
summary(model1)
```

```
##
## Call:
## lm(formula = ltrade ~ lgdp1 + lgdp2 + ldist, data = trade2013)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -14.2884  -1.1884   0.1834   1.4206  10.4898
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept) -40.97351    0.44576  -91.92  <2e-16 ***
## lgdp1        1.15138    0.01031   111.63  <2e-16 ***
## lgdp2        1.03594    0.01087    95.34  <2e-16 ***
## ldist       -1.52501    0.02756   -55.33  <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 2.317 on 11300 degrees of freedom
## (11220 observations deleted due to missingness)
## Multiple R-squared:  0.6556, Adjusted R-squared:  0.6555
## F-statistic: 7170 on 3 and 11300 DF, p-value: < 2.2e-16
```

4 推定結果の表示 (stargazer)

論文に掲載できるような表を作成する。

```
library(stargazer)
stargazer(modell, type = "text", title = "Results")
```

```
##
## Results
## =====
##                               Dependent variable:
##                               -----
##                               ltrade
## -----
## lgdp1                        1.151***
##                               (0.010)
##
## lgdp2                        1.036***
##                               (0.011)
##
## ldist                        -1.525***
##                               (0.028)
##
## Constant                    -40.974***
##                               (0.446)
##
## -----
## Observations                11,304
## R2                          0.656
## Adjusted R2                 0.655
## Residual Std. Error        2.317 (df = 11300)
## F Statistic                 7,169.780*** (df = 3; 11300)
## =====
## Note:                       *p<0.1; **p<0.05; ***p<0.01
```

推定結果を以下の式が成り立つことを示す。

$$ltrade = -40.974 + 1.151l \times gdp1 + 1.036 \times lgdp2 - 1.525 \times ldist + \text{残差}$$

2 回帰分析 (lm)

3 推定結果の表示
(summary)

4 推定結果の表示
(stargazer)

5 散布図 with plot
and abline

6 散布図 with ggplot2

7 動的な散布図 with
plotly

8 グラフの保存について

つまり、以下の関係が成り立っている。

$$\begin{aligned} \text{輸出額の対数値} = & -40.974 + 1.151 \text{輸出国のGDPの対数値} \\ & + 1.036 \text{輸入国のGDPの対数値} - 1.525 \text{距離の対数値} + \text{残差} \end{aligned}$$

5 散布図 with plot and abline

グラフの保存は、Base Rの場合、`png("figure.png")` と `dev.off()` をグラフのコードの前後に入れることでできる。

```
png("figure.png")
plot(x, y)
dev.off()
```

ただし、このようにグラフ保存を行うと、Rマークダウンにはグラフが表示されなくなってしまふ。Rマークダウンにもグラフを表示したい場合は、グラフ保存後、`` というコードをコード・チャンクの外側にかく必要がある。

```
#グラフの保存準備、解像度を100に指定
png("trade_data01.png", res = 100)

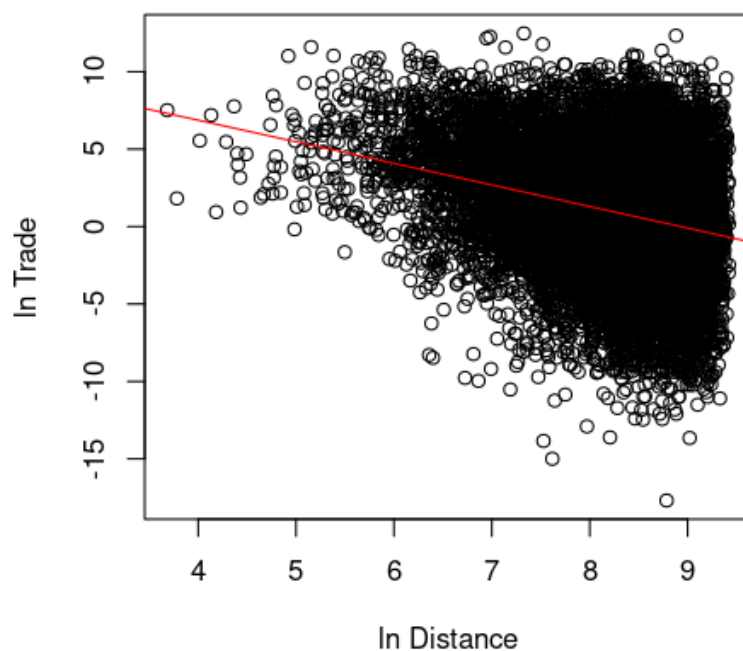
#散布図
plot(trade2013$ldist, trade2013$ltrade, xlab = "ln Distance", ylab = "ln
Trade")

#回帰直線をzに保存
z <- lm(ltrade ~ ldist, data = trade2013)

#散布図に回帰直線zを追加。
#オプションで回帰直線の色に赤(col=2でも同じ)を指定。
abline(z, col= "red")

#グラフ保存完了
dev.off()
```

```
## png
## 2
```

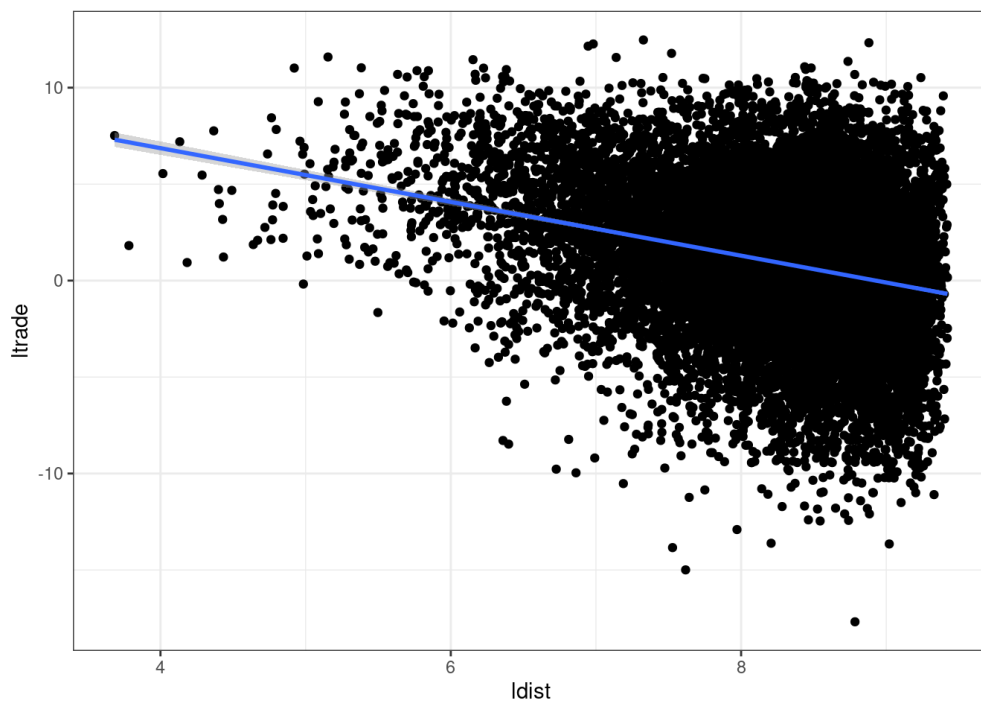


6 散布図 with ggplot2

```
library("ggplot2")

# グラフ生成
graph <- ggplot(trade2013, aes(x = ldist, y = ltrade)) +
  geom_point()+
  geom_smooth(se = TRUE, method = 'lm') +
  theme_bw()

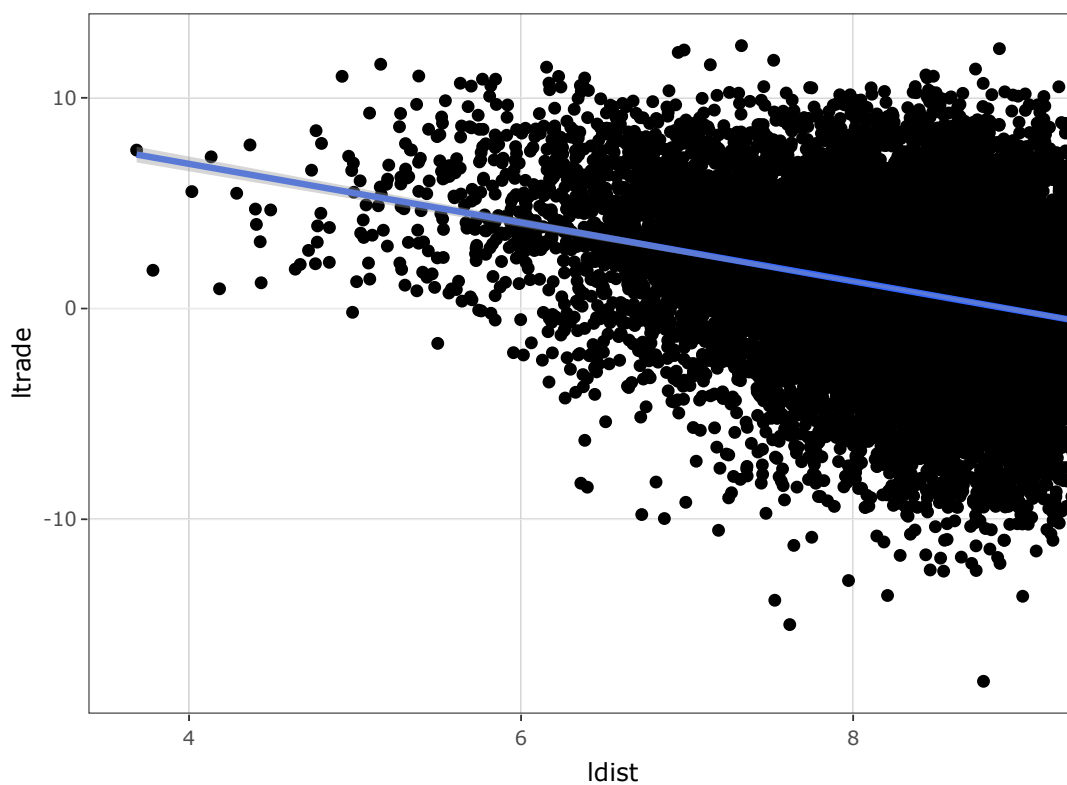
# グラフ表示
graph
```



```
# グラフ保存  
ggsave('trade_data02.png', graph, width = 5, height = 4, units = 'cm')
```

7 動的な散布図 with plotly

```
library("plotly")  
ggplotly(graph)
```



8 グラフの保存について

8.1 グラフを全部保存する場合

マークダウンの最初の YAML に、`keep_md: yes` と入れておく。

```
---
title: "貿易データの分析"
author: "田中 鮎夢"
date: "2022-12-05"
keep_md: yes #画像保存
output:
  html_document:
    toc: yes
    toc_float: yes
    number_sections: yes
---
```

さらに、最初のセットアップ・チャンクで以下のように設定しておく。

```
{r setup-chunk, include=FALSE}
knitr::opts_chunk$set(dev = c('pdf', 'png'), #画像をpng/pdfで保存
                      dpi = 100, #画像画質の設定
                      echo = TRUE, #コードを表示
                      cache = FALSE, #TRUEにするとキャッシュを残し、繰り返し実行を回避
                      message=FALSE, warning=FALSE)
```

また、コード・チャンクに `{r plotly, message=FALSE, warning=FALSE}` のように、コード・チャンクの名前を指定しておくと、画像の名前もコード・チャンクの名前に従う。

参考: Save the images that you create (<https://www.njtierney.com/post/2018/02/28/three-r-tips/>)

8.2 グラフを個別保存する場合

`ggplot2` で作成したグラフを保存する場合は、グラフ描画後、`ggsave('figure.pdf')` というコードを入れる。デフォルトでは、直前に描画したグラフが保存される。保存するグラフを指定したい場合は、以下のようにグラフ名を付与しておく。

```
グラフ名 <- ggplot(データ名, aes(x = x1, y = y1)) +
  geom_point()+
  ggsave('figure.pdf', グラフ名)
```