

# 計量経済学の課題用テンプレート

サブタイトル

氏名(学籍番号)

2024-08-19

## 1 準備

```
#dir.create("data")
```

## 2 データの整形

### 2.1 説明変数

まずはデータを読み込む.

```
data_pop_density <- read.csv("data/X/FEH_00200502_240810184536.csv",  
                             fileEncoding="cp932")
```

分析に使う変数のみを選び, 変数の名前を変更する.

```
data_pop_density_tidy <- data_pop_density|>  
  dplyr::select(統計名., X, X.8, X.30) |>  
  dplyr::rename(year = 統計名.,  
                area = X,  
                pop_dens = X.8,  
                dependency_ratio = X.30) |>
```

#area = 0 は全国を表すの  
#今回はいらないので除去する

```
subset(area!=0) |>
```

#必要な年代の変数のみを取り出す

```
dplyr::slice(-1:-998) |>
```

```
dplyr::slice(-1176:-1269)
```

data\_pop\_density には**可住地面積 1 km2 当たり人口密度【人】**と**従属人口指数**のデータが 47 都道府県 × 25(1996~2020)=1175 入っている

- 従属人口指数とは、100 人の生産年齢人口当たりの非生産年齢人口の数を表したのものである

以下、各年の総人口と生産年齢人口のデータを整形し、保存する

NA の行は地域ブロックを区切るために挿入されたものである。分析に使う変数に欠損値があったわけではない。

```
df1996 <- read_excel("data/X/1996.xls") |> #データの読み込み
dplyr::select(!c(...6,...12)) |>          #不要な変数の除去
na.omit() |>                               #NA の行を除く
dplyr::slice(-1) |>                        #最初の行は全国区
                                           #変数の改名
set_colnames(value = c("", "total_pop", "", "product_pop")) |>
dplyr::select("total_pop", "product_pop")  #変数の選択
```

```
df1997 <- read_excel("data/X/1997.xls") |>      #データの読み込み
na.omit() |>                                     #NA の行を消す
                                           #変数の改名
set_colnames(value = c("", "", "total_pop", "", "product_pop", "")) |>
dplyr::select("total_pop", "product_pop") #変数の選択
```

```
#データの読み込み
df1998 <- read_excel("data/X/1998.xls") |>
#NA の行の z 除去
na.omit() |>
#変数の改名
set_colnames(value = c("", "", "total_pop", "", "product_pop")) |>
#必要な変数の選択
dplyr::select("total_pop", "product_pop",)
```

```
#データの読み込み
df1999 <- read_excel("data/X/1999.xls") |>
#NA の行の削除
na.omit() |>
#変数の改名
set_colnames(value = c("NA" , "NA" , "total_pop", "NA" , "product_pop"))|>
#必要な変数の選択
```

```
dplyr::select("total_pop", "product_pop")
```

#データの読み込み

```
df2000 <- read_excel("data/X/2000.xlsx") |>  
  #必要な変数の選択  
  dplyr::select(...6, c(...10:...19)) |>  
  #NA の行の削除  
  na.omit() |>  
  #不要な列の削除  
  dplyr::slice(c(-1:-2), c(-50:-147)) |>  
  #変数の改名  
  set_colnames(value = c("total_pop", "a", "b", "c", "d", "e", "f", "g", "h", "i", "j")) |>  
  #変数を numeric に変換  
  mutate(across(a:j, as.numeric)) |>  
  #product_pop を作る (ないので)  
  mutate(product_pop = rowSums(across(a:j, ~.))) |>  
  #必要な変数の選択  
  dplyr::select("total_pop", "product_pop")
```

#データの読み込み

```
df2001 <- read_excel("data/X/2001.xls") |>  
  #NA の行の除去  
  na.omit() |>  
  #変数の改名  
  set_colnames(value = c("NA" , "NA" , "total_pop", "NA" , "product_pop")) |>  
  #必要な変数の選択  
  dplyr::select("total_pop", "product_pop")
```

#データの読み込む

```
df2002 <- read_excel("data/X/2002.xls") |>  
  #NA の行の削除  
  na.omit() |>  
  #変数の改名  
  set_colnames(value = c("NA" , "NA" , "total_pop",  
                        "NA" , "product_pop")) |>  
  #必要な変数の選択  
  dplyr::select("total_pop", "product_pop")
```

#データの読むこみ

```
df2003 <- read_excel("data/X/2001.xls") |>
```

```
#NA の行を削除
na.omit() |>
#変数の改名
set_colnames(value = c("NA" , "NA" ,"total_pop", "NA" ,"product_pop"))|>
#必要な変数の除去
dplyr::select("total_pop","product_pop")
```

```
#データの読み込み
df2004 <- read_excel("data/X/2001.xls") |>
#NA の行の削除
na.omit() |>
#変数の改名
set_colnames(value = c("NA" , "NA" ,"total_pop", "NA" ,"product_pop"))|>
#必要な変数の選択
dplyr::select("total_pop","product_pop")
```

```
#データの読み込み
df2005 <- read_excel("data/X/2000.xlsx")|>
#product_pop を作るのに必要な変数を選ぶ
dplyr::select(...6,c(...10:...19)) |>
#NA の行の削除
na.omit() |>
#不要な列の削除
dplyr::slice(c(-1:-2),c(-50:-147)) |>
#変数の改名
set_colnames(value = c("total_pop","a","b","c","d","e","f","g","h","i","j")) |>
#変数を numeric に変換
mutate(across(a:j,as.numeric)) |>
#product_pop を作る
mutate(product_pop = rowSums(across(a:j,~.))) |>
#必要な変数の選択
dplyr::select("total_pop","product_pop")
```

```
#データの読み込み
df2006 <- read_excel("data/X/2001.xls") |>
#NA の行の除去
na.omit() |>
#変数の改名
set_colnames(value = c("NA" , "NA" ,"total_pop", "NA" ,"product_pop"))|>
#必要な変数の選択
```

```
dplyr::select("total_pop", "product_pop")
```

#データの読み込み

```
df2007 <- read_excel("data/X/2007.xlsx") |>  
  #必要な変数の選択  
  dplyr::select("1", "3") |>  
  #不要な列の削除  
  dplyr::slice(c(-1:-19)) |>  
  #必要な変数の選択  
  set_colnames(value = c("total_pop", "product_pop"))
```

#データの読み込み

```
df2008 <- read_excel("data/X/2008.xls") |>  
  #必要な変数のみを選択  
  dplyr::select("1", "3") |>  
  #不要な行の削除  
  dplyr::slice(c(-1:-19)) |>  
  #変数の改名  
  set_colnames(value = c("total_pop", "product_pop"))
```

#データの読み込み

```
df2009 <- read_excel("data/X/2009.xls") |>  
  #必要な変数の選択  
  dplyr::select("1", "3") |>  
  #不要な行の削除  
  dplyr::slice(c(-1:-19)) |>  
  #変数の改名  
  set_colnames(value = c("total_pop", "product_pop"))
```

#データの読み込み

```
df2010 <- read_excel("data/X/2000.xlsx") |>  
  #product_pop の計算に必要な変数の選択  
  dplyr::select(...6, c(...10:...19)) |>  
  #NA の行の削除  
  na.omit() |>  
  #不要な行の削除  
  dplyr::slice(c(-1:-2), c(-50:-147)) |>  
  #変数の改名  
  set_colnames(value = c("total_pop", "a", "b", "c", "d", "e", "f", "g", "h", "i", "j")) |>  
  #変数を as.numeric に変換
```

```
mutate(across(a:j,as.numeric)) |>
#product_pop の作成
mutate(product_pop = rowSums(across(a:j,~.))) |>
#必要な変数の選択
dplyr::select("total_pop","product_pop")
```

#データの読み込み

```
df2011 <- read_excel("data/X/2011.xls") |>
#必要な変数の選択
dplyr::select("1","3") |>
#不要な列の削除
dplyr::slice(c(-1:-19)) |>
#変数の改名
set_colnames(value = c("total_pop","product_pop"))
```

#データの読み込み

```
df2012 <- read_excel("data/X/2012.xls") |>
#必要な変数の選択
dplyr::select("1","3") |>
#不要な列の削除
dplyr::slice(c(-1:-19)) |>
#変数の改名
set_colnames(value = c("total_pop","product_pop"))
```

#データの読み込み

```
df2013 <- read_excel("data/X/2013.xls") |>
#必要な変数の選択
dplyr::select("1","3") |>
#不要な行の削除
dplyr::slice(c(-1:-19)) |>
#変数の改名
set_colnames(value = c("total_pop","product_pop"))
```

#データの読み込み

```
df2014 <- read_excel("data/X/2014.xls") |>
#必要な変数の選択
dplyr::select("1","3") |>
#不要な行の削除
dplyr::slice(c(-1:-19)) |>
#変数の改名
```

```
set_colnames(value = c("total_pop", "product_pop"))
```

#データの読み込む

```
df2015 <- read_excel("data/X/2000.xlsx") |>
  #product_pop の計算に必要な変数の選択
  dplyr::select(...6, c(...10:...19)) |>
  #NA の行の削除
  na.omit() |>
  #不要な行の削除
  dplyr::slice(c(-1:-2), c(-50:-147)) |>
  #変数の改名
  set_colnames(value = c("total_pop", "a", "b", "c", "d", "e", "f", "g", "h", "i", "j")) |>
  #変数を numeric にする
  mutate(across(a:j, as.numeric)) |>
  #product_pop を計算する
  mutate(product_pop = rowSums(across(a:j, ~.))) |>
  #必要な変数の選択
  dplyr::select("total_pop", "product_pop")
```

#データの読み込み

```
df2016 <- read_excel("data/X/2016.xls") |>
  #必要な変数の選択
  dplyr::select("1", "3") |>
  #不要な行の削除
  dplyr::slice(c(-1:-22), c(-48:-117)) |>
  #変数の改名
  set_colnames(value = c("total_pop", "product_pop"))
```

#データの読み込み

```
df2017 <- read_excel("data/X/2017.xls") |>
  #必要な変数の選択
  dplyr::select("1", "3") |>
  #不要な行の削除
  dplyr::slice(c(-1:-22), c(-48:-117)) |>
  #変数の改名
  set_colnames(value = c("total_pop", "product_pop"))
```

#データの読み込み

```
df2018 <- read_excel("data/X/2018.xls") |>
  #必要な変数の選択
```

```
dplyr::select("1","3") |>
#不要な行の削除
dplyr::slice(c(-1:-22),c(-48:-117)) |>
#変数の改名
set_colnames(value = c("total_pop","product_pop"))
```

#データの読み込み

```
df2019 <- read_excel("data/X/2019.xls") |>
#必要な変数の選択
dplyr::select("1","3") |>
#不要な行の削除
dplyr::slice(c(-1:-22),c(-48:-117)) |>
#変数の改名
set_colnames(value = c("total_pop","product_pop"))
```

#データの読み込み

```
df2020 <- read_excel("data/X/2000.xlsx") |>
#product_pop の計算に必要な変数の選択
dplyr::select(...6,c(...10:...19)) |>
#NA の行の削除
na.omit() |>
#NA の行の削除
dplyr::slice(c(-1:-2),c(-50:-147)) |>
#変数の改名
set_colnames(value = c("total_pop","a","b","c","d","e","f","g","h","i","j")) |>
#変数を numeric に変換する
mutate(across(a:j,as.numeric)) |>
#product_pop を計算する
mutate(product_pop = rowSums(across(a:j,~.))) |>
#必要な変数の選択
dplyr::select("total_pop","product_pop")
```

最後に、各年のデータを結合する。

```
df <- rbind(
  df1996,df1997,df1998,df1999,df2000,
  df2001,df2002,df2003,df2004,df2005,
  df2006,df2007,df2008,df2009,df2010,
  df2011,df2012,df2013,df2014,df2015,
  df2016,df2017,df2018,df2019,df2020)
```



可住面積あたりの人口密度と従属人口指数が入ったデータと総人口と生産年齢人口が入ったデータを結合し,説明変数にしようするデータが揃った.

```
data_X<- cbind(data_pop_density_tidy,df)
```

## 2.2 応答変数

以下,データセットから正式系列とされている年のデータのみを抽出し,調査年ごとに df に格納する

```
#データの読み込み
df_income_1 <- read_excel("data/Y/one_income_1.xlsx")

df_gdp_1 <- read_excel("data/Y/total_gdp_1.xlsx")

#不要な行を消す
df_income_1 <- df_income_1 |>
  dplyr::slice(c(-1:-5),c(-53:-78)) |>
  set_colnames(c("", "", "", "2011", "2012",
                  "2013", "2014", "2015", "2016", "2017",
                  "2018", "2019", "2020"))

df_gdp_1 <- df_gdp_1 |>
  dplyr::slice(c(-1:-5),c(-53:-78)) |>
  set_colnames(c("", "", "", "2011", "2012",
                  "2013", "2014", "2015", "2016", "2017",
                  "2018", "2019", "2020"))

#年観測年ごとにデータをまとめる
y2020 <- cbind(df_income_1$"2020",df_gdp_1$"2020")
y2019 <- cbind(df_income_1$"2019",df_gdp_1$"2019")
y2018 <- cbind(df_income_1$"2018",df_gdp_1$"2018")
y2017 <- cbind(df_income_1$"2017",df_gdp_1$"2017")
y2016 <- cbind(df_income_1$"2016",df_gdp_1$"2016")
y2015 <- cbind(df_income_1$"2015",df_gdp_1$"2015")
y2014 <- cbind(df_income_1$"2014",df_gdp_1$"2014")
y2013 <- cbind(df_income_1$"2013",df_gdp_1$"2013")
y2012 <- cbind(df_income_1$"2012",df_gdp_1$"2012")
y2011 <- cbind(df_income_1$"2011",df_gdp_1$"2011")
```

```

#データの読み込み
df_income_2 <- read_excel("data/Y/one_income_2.xlsx")

df_gdp_2 <- read_excel("data/Y/total_gdp_2.xlsx")

#不要な行の削除
df_income_2 <- df_income_2 |>
  dplyr::slice(c(-1:-5),c(-53:-79)) |>
  set_colnames(c("", "", "", "2006", "2007",
                  "2008", "2009", "2010", "2011", "2012",
                  "2013", "2014", "2015", "2016", "2017", "2018"))

df_gdp_2 <- df_gdp_2 |>
  dplyr::slice(c(-1:-5),c(-53:-79)) |>
  set_colnames(c("", "", "", "2006", "2007",
                  "2008", "2009", "2010", "2011", "2012",
                  "2013", "2014", "2015", "2016", "2017", "2018"))

#観測年ごとにデータを格納する
y2010 <- cbind(df_income_2$"2010",df_gdp_2$"2010")
y2009 <- cbind(df_income_2$"2009",df_gdp_2$"2009")
y2008 <- cbind(df_income_2$"2008",df_gdp_2$"2008")
y2007 <- cbind(df_income_2$"2007",df_gdp_2$"2007")
y2006 <- cbind(df_income_2$"2006",df_gdp_2$"2006")

```

```

#データの読み込み
df_income_3 <- read_excel("data/Y/one_income_3.xls")

df_gdp_3 <- read_excel("data/Y/total_gdp_3.xls")

#不要な行の削除
df_income_3 <- df_income_3 |>
  dplyr::slice(c(-1:-5),c(-53:-79)) |>
  set_colnames(c("", "", "", "2001", "2002",
                  "2003", "2004", "2005", "2006", "2007",
                  "2008", "2009", "2010", "2011", "2012", "2013", "2014"))

df_gdp_3 <- df_gdp_3 |>
  dplyr::slice(c(-1:-5),c(-53:-79)) |>

```

```

set_colnames(c("", "", "", "2001", "2002",
               "2003", "2004", "2005", "2006", "2007",
               "2008", "2009", "2010", "2011", "2012", "2013", "2014"))

#観測年ごとにデータを格納する
y2005 <- cbind(df_income_3$"2005", df_gdp_3$"2005")
y2004 <- cbind(df_income_3$"2004", df_gdp_3$"2004")
y2003 <- cbind(df_income_3$"2003", df_gdp_3$"2003")
y2002 <- cbind(df_income_3$"2002", df_gdp_3$"2002")
y2001 <- cbind(df_income_3$"2001", df_gdp_3$"2001")

```

```

#データの読み込み
df_income_4 <- read_excel("data/Y/one_income_4.xls")

df_gdp_4 <- read_excel("data/Y/total_gdp_4.xls")

#不要な行の削除
df_income_4 <- df_income_4 |>
  dplyr::slice(c(-1:-5), c(-53:-75)) |>
  set_colnames(c("", "", "", "1996", "1997",
                  "1998", "1999", "2000", "2001", "2002",
                  "2003", "2004", "2005", "2006", "2007", "2008", "2009"))

df_gdp_4 <- df_gdp_4 |>
  dplyr::slice(c(-1:-5), c(-53:-75)) |>
  set_colnames(c("", "", "", "1996", "1997",
                  "1998", "1999", "2000", "2001", "2002",
                  "2003", "2004", "2005", "2006", "2007", "2008", "2009"))

```

```

#観測年ごとにデータを格納する
y2000 <- cbind(df_income_4$"2000", df_gdp_4$"2000")
y1999 <- cbind(df_income_4$"1999", df_gdp_4$"1999")
y1998 <- cbind(df_income_4$"1998", df_gdp_4$"1998")
y1997 <- cbind(df_income_4$"1997", df_gdp_4$"1997")
y1996 <- cbind(df_income_4$"1996", df_gdp_4$"1996")

```

全ての調査年の df が揃ったので結合する. 変数名の変換も同時に行っておく.

```
data_y <- rbind(
  y1996,y1997,y1998,y1999,y2000,
  y2001,y2002,y2003,y2004,y2005,
  y2006,y2007,y2008,y2009,y2010,
  y2011,y2012,y2013,y2014,y2015,
  y2016,y2017,y2018,y2019,y2020)

data_y <- data.frame(data_y) |>
  rename(income = X1,
         gdp = X2)
```

### 3 モデル

固定効果 (  $\gamma_i$  ), 時間効果 (  $\lambda_t$  ), および交差項を含むモデルは、次のように表現できる

$$Y_{it} = \beta_1 X_{it} + \beta_2 Z_{it} + \beta_3 (X_{it} \times Z_{it}) + \alpha_i + \lambda_t + u_{it}$$

ここで,

- $Y_{it}$  は個体  $i$  の時点  $t$  における応答変数.
- $X_{it}$  と  $Z_{it}$  は説明変数.
- $(X_{it} \times Z_{it})$  は交差項.
- $\alpha_i$  は都道府県効果.
- $\lambda_t$  は時間効果.
- $u_{it}$  は誤差項.

また,

- $Y_{it}$  は経済状況を表す変数
- $X_{it}$  は生産年齢人口を表す変数
- $Z_{it}$  は人口密度を表す変数

都道府県効果と時間効果を除去するための変数変換は以下の通りである.

$$\dot{Y}_{it} = Y_{it} - \bar{Y}_i - \bar{Y}_t + \bar{Y}$$

$$\begin{aligned}
&= [\beta_1 X_{it} + \beta_2 Z_{it} + \beta_3 (X_{it} \times Z_{it}) + \alpha_i + \lambda_t + u_{it}] \\
&\quad - \frac{1}{T} \sum_{t=1}^T [\beta_1 X_{it} + \beta_2 Z_{it} + \beta_3 (X_{it} \times Z_{it}) + \alpha_i + \lambda_t + u_{it}] \\
&\quad - \frac{1}{N} \sum_{i=1}^N [\beta_1 X_{it} + \beta_2 Z_{it} + \beta_3 (X_{it} \times Z_{it}) + \alpha_i + \lambda_t + u_{it}] \\
&\quad + \frac{1}{TN} \sum_{t=1}^T \sum_{i=1}^N [\beta_1 X_{it} + \beta_2 Z_{it} + \beta_3 (X_{it} \times Z_{it}) + \alpha_i + \lambda_t + u_{it}] \\
&= [\beta_1 X_{it} + \beta_2 Z_{it} + \beta_3 (X_{it} \times Z_{it}) + \alpha_i + \lambda_t + u_{it}] \\
&\quad - [\beta_1 \frac{1}{T} \sum_{t=1}^T X_{it} + \beta_2 \frac{1}{T} \sum_{t=1}^T Z_{it} + \beta_3 \frac{1}{T} \sum_{t=1}^T (X_{it} \times Z_{it}) + \frac{1}{T} \sum_{t=1}^T \alpha_i + \frac{1}{T} \sum_{t=1}^T \lambda_t + \frac{1}{T} \sum_{t=1}^T u_{it}] \\
&\quad - [\beta_1 \frac{1}{N} \sum_{i=1}^N X_{it} + \beta_2 \frac{1}{N} \sum_{i=1}^N Z_{it} + \beta_3 \frac{1}{N} \sum_{i=1}^N (X_{it} \times Z_{it}) + \frac{1}{N} \sum_{i=1}^N \alpha_i + \frac{1}{N} \sum_{i=1}^N \lambda_t + \frac{1}{N} \sum_{i=1}^N u_{it}] \\
&\quad + [\beta_1 \frac{1}{TN} \sum_{t=1}^T \sum_{i=1}^N X_{it} + \beta_2 \frac{1}{TN} \sum_{t=1}^T \sum_{i=1}^N Z_{it} + \beta_3 \frac{1}{TN} \sum_{t=1}^T \sum_{i=1}^N (X_{it} \times Z_{it}) \\
&\quad + \frac{1}{TN} \sum_{t=1}^T \sum_{i=1}^N \alpha_i + \frac{1}{TN} \sum_{t=1}^T \sum_{i=1}^N \lambda_t + \frac{1}{TN} \sum_{t=1}^T \sum_{i=1}^N u_{it}] \\
&= [\beta_1 X_{it} + \beta_2 Z_{it} + \beta_3 (X_{it} \times Z_{it}) + \alpha_i + \lambda_t + u_{it}] \\
&\quad - [\beta_1 \bar{X}_i + \beta_2 \bar{Z}_i + \beta_3 \overline{(X_i \times Z_i)} + \alpha_i + \bar{\lambda} + \bar{u}_i] \\
&\quad - [\beta_1 \bar{X}_t + \beta_2 \bar{Z}_t + \beta_3 \overline{(X_t \times Z_t)} + \bar{\alpha} + \lambda_t + \bar{u}_t] \\
&\quad + [\beta_1 \bar{X} + \beta_2 \bar{Z} + \beta_3 \overline{(X \times Z)} + \bar{\alpha} + \bar{\lambda} + \bar{u}] \\
&= \beta_1 [X_{it} - \bar{X}_i - \bar{X}_t + \bar{X}] \\
&\quad + \beta_2 [Z_{it} - \bar{Z}_i - \bar{Z}_t + \bar{Z}] \\
&\quad + \beta_3 [(X_{it} \times Z_{it}) - \overline{(X_i \times Z_i)} - \overline{(X_t \times Z_t)} + \overline{(X \times Z)}] \\
&\quad + [\alpha_i - \alpha_i - \bar{\alpha} + \bar{\alpha}] \\
&\quad + [\lambda_t - \bar{\lambda} - \lambda_t + \bar{\lambda}] \\
&\quad + [u_{it} - \bar{u}_i - \bar{u}_t + \bar{u}] \\
&= \beta_1 \dot{X}_{it} + \beta_2 \dot{Z}_{it} + \beta_3 (\dot{X}_{it} \times \dot{Z}_{it}) + \dot{u}_{it}
\end{aligned}$$

時間効果として考えられるものは、景気の変動である。所得や GDP は国の経済政策の効果に依存すること t は容易に想像できる。また、国が行う政策で特定の地域が優遇されるということも考えにくい。

都道府県効果として考えられるものは、各都道府県の自治体を実施する経済政策や産業構造である。

### 3.1 単位の確認

今後の分析に使うものとして、総人口に占める生産年齢人口の割合の変数 `pop_bon` と従属人口指数の逆数をとった変数 `ratio_rec` を作った。

```
#必要なデータの結合
data <- cbind(data_X,data_y) |>
#カンマの削除
mutate(across(year:gdp,~gsub(",","",.))) |>
#変数を numeric に変換する
mutate(across(year:gdp,as.numeric)) |>
mutate(pop_bon = product_pop/total_pop * 100,
       ratio_rec = 1/(dependency_ratio))
```

変数	説明	単位
year	調査された年	一年
area	調査対象の場所	都道府県
pop_dens	可住地面積 1 km2 当たり人口密度	人
dependency_ratio	従属人口指数	(%)
total_pop	各年, 各都道府県の総人口	千人
product_pop	各年, 各都道府県の生産年齢人口	千人
income	各年, 各都道府県の一人当たり所得	千円
gdp	各年, 各都道府県の名目 GDP	百万円
pop_bon	総人口に占める生産年齢人口の比	(%)
ratio_rec	従属人口指数の逆数	なし

可住地面積 1 km2 当たり人口密度の従属人口指数のデータの出典は [e-Stat](#) である

総人口と生産年齢人口のデータの出典は総務省統計局の[各年 10 月 1 日現在人口](#)である。

gdp と所得のデータの出典は内閣府の[統計表 \(県民経済計算\)](#) である

gdp と所得は正式系列のものを採用している

詳細は内閣府の県民経済計算の[利用場の注意](#)を参照

### 3.2 変数変換

data の中身を確認する

```
summary(data)
```

```

      year          area      pop_dens      dependency_ratio
Min.   :1.996e+09   Min.    : 1000   Min.    : 230.2   Min.    :35.70
1st Qu.:2.002e+09   1st Qu.:12000   1st Qu.: 633.5   1st Qu.:53.50
Median :2.008e+09   Median :24000   Median : 841.4   Median :60.00
Mean   :2.008e+09   Mean    :24000   Mean    :1367.3   Mean    :60.76
3rd Qu.:2.014e+09   3rd Qu.:36000   3rd Qu.:1245.3   3rd Qu.:67.65
Max.   :2.020e+09   Max.    :47000   Max.    :9873.3   Max.    :89.90

      total_pop      product_pop      income      gdp
Min.    : 551   Min.    : 304   Min.    :1933   Min.    : 1725231
1st Qu.: 1168   1st Qu.: 726   1st Qu.:2538   1st Qu.: 3845666
Median : 1743   Median :1022   Median :2791   Median : 5900703
Mean    : 2700   Mean    :1767   Mean    :2825   Mean    :11412941
3rd Qu.: 2844   3rd Qu.:1816   3rd Qu.:3008   3rd Qu.:11019084
Max.    :13921   Max.    :9158   Max.    :5970   Max.    :115063263

      pop_bon      ration_rec
Min.    :52.93   Min.    :0.01112
1st Qu.:60.65   1st Qu.:0.01478
Median :63.55   Median :0.01667
Mean    :63.42   Mean    :0.01698
3rd Qu.:66.33   3rd Qu.:0.01869
Max.    :73.70   Max.    :0.02801

```

pop\_dens, income, gdp の値が他の変数に比べて大きすぎる. 分析結果の解釈をしやすくするために, 単位変換を行う. また, year と area の値の後ろに 0 が余計についていたため, 取り除く.

```

data <- data |>
  mutate(gdp = gdp / 10^4,
         income = income / 10,
         pop_dens = pop_dens / 10^3,
         year = (year / 10^6 )-0.1,
         area = area / 10^3)
summary(data)

```

```

      year          area      pop_dens      dependency_ratio      total_pop
Min.   :1996   Min.    : 1   Min.    :0.2302   Min.    :35.70   Min.    : 551
1st Qu.:2002   1st Qu.:12   1st Qu.:0.6334   1st Qu.:53.50   1st Qu.: 1168
Median :2008   Median :24   Median :0.8414   Median :60.00   Median : 1743
Mean    :2008   Mean    :24   Mean    :1.3673   Mean    :60.76   Mean    : 2700
3rd Qu.:2014   3rd Qu.:36   3rd Qu.:1.2453   3rd Qu.:67.65   3rd Qu.: 2844

```

```

Max.      :2020   Max.      :47   Max.      :9.8733   Max.      :89.90   Max.      :13921
  product_pop      income      gdp      pop_bon
Min.      : 304   Min.      :193.3   Min.      : 172.5   Min.      :52.93
1st Qu.: 726   1st Qu.:253.8   1st Qu.: 384.6   1st Qu.:60.65
Median :1022   Median :279.1   Median : 590.1   Median :63.55
Mean    :1767   Mean    :282.5   Mean    :1141.3   Mean    :63.42
3rd Qu.:1816   3rd Qu.:300.8   3rd Qu.:1101.9   3rd Qu.:66.33
Max.     :9158   Max.     :597.0   Max.     :11506.3   Max.     :73.70

  ration_rec
Min.      :0.01112
1st Qu.:0.01478
Median :0.01667
Mean    :0.01698
3rd Qu.:0.01869
Max.     :0.02801

```

ここで, 変数の単位を改めて確認する

変数	説明	単位
year	調査された年	一年
area	調査対象の場所	都道府県
pop_dens	可住地面積 1 km <sup>2</sup> 当たり人口密度	千人
dependency_ratio	従属人口指数	(%)
total_pop	各年, 各都道府県の総人口	千人
product_pop	各年, 各都道府県の生産年齢人口	千人
income	各年, 各都道府県の一人当たり所得	一万円
gdp	各年, 各都道府県の名目 GDP	百億円
pop_bon	総人口に占める生産年齢人口の比	(%)
ratio_rec	従属人口指数の逆数	なし

変数変換を行う前に, 交差項の変数を作っておく

```

data <- data |>
  mutate(pop_bon_X_pop_dens = pop_bon * pop_dens,
         ratio_rec_X_pop_dens = ratio_rec * pop_dens)

```

全平均

```
all_mean <- data.frame(t(colMeans(data)))
```

t の平均



```

t_mean <- matrix(NA,nrow = 47,ncol = 12)
for(i in 1:47){
  a <- data |>
    filter(area == i)
  t_mean[i,] <- t(colMeans(a))
}
t_mean <- t_mean |>
  set_colnames(value = c(names(data)))
t_mean <- data.frame(t_mean)

```

i の平均

```

i_mean <- matrix(NA,nrow = 25,ncol = 12)
for(t in 1:25){
  a <- data |>
    filter(year == t+1995)
  i_mean[t,] <- t(colMeans(a))
}
i_mean <- i_mean |>
  set_colnames(value = c(names(data)))
i_mean <- data.frame(i_mean)

```

変数変換

```

trans <- data.frame(matrix(NA,nrow = 1175,ncol = 12))
r <- 0
for(t in 1:25){
  for(i in 1:47){
    a <- data |>
      filter(area == i,
        year == t+1995)

    b <- t_mean |>
      filter(area == i)

    c <- i_mean |>
      filter(year == t+1995)

    d <- all_mean

    x <- a - b - c + d

```

```

    r <- r + 1

    trans[r,] <- x
  }
}
trans <- trans |>
  set_colnames(value = names(data))

```

変数変換後の統計量の確認

```
summary(trans)
```

year	area	pop_dens	dependency_ratio
Min. :0	Min. :0	Min. :-0.655982	Min. :-8.27949
1st Qu.:0	1st Qu.:0	1st Qu.: -0.016221	1st Qu.: -0.88889
Median :0	Median :0	Median :-0.001746	Median :-0.06587
Mean :0	Mean :0	Mean : 0.000000	Mean : 0.00000
3rd Qu.:0	3rd Qu.:0	3rd Qu.: 0.016671	3rd Qu.: 0.91072
Max. :0	Max. :0	Max. : 0.674907	Max. : 6.55821

total_pop	product_pop	income	gdp
Min. :-805.839	Min. :-538.133	Min. :-100.8918	Min. :-1452.428
1st Qu.: -34.932	1st Qu.: -28.461	1st Qu.: -6.1330	1st Qu.: -35.022
Median : 9.885	Median : -1.541	Median : -0.9281	Median : 5.081
Mean : 0.000	Mean : 0.000	Mean : 0.0000	Mean : 0.000
3rd Qu.: 34.603	3rd Qu.: 23.547	3rd Qu.: 5.7200	3rd Qu.: 36.465
Max. :1350.289	Max. : 558.147	Max. : 72.5116	Max. : 1305.901

pop_bon	ration_rec	pop_bon_X_pop_dens
Min. :-2.245402	Min. :-2.040e-03	Min. :-41.98903
1st Qu.: -0.392750	1st Qu.: -3.744e-04	1st Qu.: -1.68753
Median : 0.007484	Median : 1.813e-05	Median : -0.09385
Mean : 0.000000	Mean : 0.000e+00	Mean : 0.00000
3rd Qu.: 0.387873	3rd Qu.: 3.490e-04	3rd Qu.: 1.64900
Max. : 2.553979	Max. : 3.272e-03	Max. : 68.17836

ration_rec_X_pop_dens
Min. :-0.0251705
1st Qu.: -0.0018104
Median : 0.0002525
Mean : 0.0000000
3rd Qu.: 0.0016848
Max. : 0.0415356

## 4 分析

### 4.1 分析の実行

lm 関数を使って時間効果と都道府県効果と交差項を含んだモデルの分析を行う

- モデル 1

$\dot{Y}_{it} = \beta_1 \dot{X}_{it} + \beta_2 \dot{Z}_{it} + \beta_3 (\dot{X}_{it} \times \dot{Z}_{it}) + \dot{u}_{it}$  ただし,  $Y_{it}$  は一人当たり所得,  $X_{it}$  は人口ボーナス度,  $Z_{it}$  は人口密度である

```
model_1 <- lm(income ~ 0+pop_bon + pop_dens + pop_bon_X_pop_dens, data = trans)
summary(model_1)
```

Call:

```
lm(formula = income ~ 0 + pop_bon + pop_dens + pop_bon_X_pop_dens,
    data = trans)
```

Residuals:

Min	1Q	Median	3Q	Max
-63.845	-6.243	-0.660	5.573	77.769

Coefficients:

	Estimate	Std. Error	t value	Pr(> t )
pop_bon	1.44278	0.57071	2.528	0.011601 *
pop_dens	50.52795	4.93304	10.243	< 2e-16 ***
pop_bon_X_pop_dens	0.25190	0.06918	3.641	0.000283 ***

---

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

Residual standard error: 11.16 on 1172 degrees of freedom

Multiple R-squared: 0.1522, Adjusted R-squared: 0.15

F-statistic: 70.11 on 3 and 1172 DF, p-value: < 2.2e-16

- モデル 2

$\dot{Y}_{it} = \beta_1 \dot{X}_{it} + \beta_2 \dot{Z}_{it} + \beta_3 (\dot{X}_{it} \times \dot{Z}_{it}) + \dot{u}_{it}$  ただし,  $Y_{it}$  は一人当たり所得,  $X_{it}$  は従属人口指数の逆数,  $Z_{it}$  は人口密度である

```
model_2 <- lm(income ~ 0+ration_rec + pop_dens + ration_rec_X_pop_dens, data = trans)
summary(model_2)
```

Call:

```
lm(formula = income ~ 0 + ration_rec + pop_dens + ration_rec_X_pop_dens,  
    data = trans)
```

Residuals:

Min	1Q	Median	3Q	Max
-66.590	-6.158	-0.939	5.643	83.517

Coefficients:

	Estimate	Std. Error	t value	Pr(> t )
ration_rec	3106.864	623.175	4.986	7.11e-07 ***
pop_dens	79.400	4.997	15.890	< 2e-16 ***
ration_rec_X_pop_dens	391.138	100.030	3.910	9.75e-05 ***

---

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

Residual standard error: 10.79 on 1172 degrees of freedom

Multiple R-squared: 0.2073, Adjusted R-squared: 0.2053

F-statistic: 102.2 on 3 and 1172 DF, p-value: < 2.2e-16

- モデル3

$\dot{Y}_{it} = \beta_1 \dot{X}_{it} + \beta_2 \dot{Z}_{it} + \beta_3 (\dot{X}_{it} \times \dot{Z}_{it}) + \dot{u}_{it}$  ただし、 $Y_{it}$  は県内総生産額、 $X_{it}$  は人口ボーナス度、 $Z_{it}$  は人口密度である

```
model_3 <- lm(gdp ~ 0 + pop_bon + pop_dens + pop_bon_X_pop_dens, data = trans)  
summary(model_3)
```

Call:

```
lm(formula = gdp ~ 0 + pop_bon + pop_dens + pop_bon_X_pop_dens,  
    data = trans)
```

Residuals:

Min	1Q	Median	3Q	Max
-585.86	-28.01	1.78	26.89	560.74

Coefficients:

	Estimate	Std. Error	t value	Pr(> t )
pop_bon	10.1843	3.8684	2.633	0.00858 **

```
pop_dens          1572.9319    33.4366  47.042 < 2e-16 ***
pop_bon_X_pop_dens -1.5178      0.4689  -3.237  0.00124 **
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

Residual standard error: 75.65 on 1172 degrees of freedom  
Multiple R-squared: 0.6828, Adjusted R-squared: 0.682  
F-statistic: 841 on 3 and 1172 DF, p-value: < 2.2e-16

- モデル 4

$\dot{Y}_{it} = \beta_1 \dot{X}_{it} + \beta_2 \dot{Z}_{it} + \beta_3 (\dot{X}_{it} \times \dot{Z}_{it}) + \dot{u}_{it}$  ただし,  $Y_{it}$  は県内総生産額,  $X_{it}$  は従属人口指数の逆数,  $Z_{it}$  は人口密度である

```
model_4 <- lm(gdp ~ 0+ration_rec + pop_dens + ration_rec_X_pop_dens, data = trans)
summary(model_4)
```

Call:

```
lm(formula = gdp ~ 0 + ration_rec + pop_dens + ration_rec_X_pop_dens,
    data = trans)
```

Residuals:

Min	1Q	Median	3Q	Max
-627.47	-27.92	1.20	26.38	544.28

Coefficients:

	Estimate	Std. Error	t value	Pr(> t )
ration_rec	3875.17	4388.19	0.883	0.377
pop_dens	1518.42	35.19	43.153	<2e-16 ***
ration_rec_X_pop_dens	-677.87	704.38	-0.962	0.336

---

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

Residual standard error: 76 on 1172 degrees of freedom  
Multiple R-squared: 0.6799, Adjusted R-squared: 0.6791  
F-statistic: 829.9 on 3 and 1172 DF, p-value: < 2.2e-16

## 4.2 結果の可視化

- 係数を表示する

表 3: 説明変数が生産年齢人口

	一人当たり所得 (一万円)	県内総生産額 (百億円)
生産年齢人口の割合 (%)	1.443* (0.571)	10.184** (3.868)
人口密度 (千人/km <sup>2</sup> )	50.528*** (4.933)	1572.932*** (33.437)
生産年齢人口の割合 x 人口密度	0.252*** (0.069)	-1.518** (0.469)
R <sup>2</sup>	0.152	0.683
自由度決定済み R <sup>2</sup>	0.150	0.682
観測数	1175	1175

注: 括弧内は標準偏差

```
res <- list(model_1,model_3)
texreg(res,
  stars = c(0.001, 0.01, 0.05),
  digits = 3,
  caption = "説明変数が生産年齢人口",
  caption.above = TRUE,
  custom.model.names = c("一人当たり所得 (一万円)", "県内総生産額 (百億円)"),
  custom.coef.name = c("生産年齢人口の割合 (%)",
    "人口密度 (千人/km^2)",
    "生産年齢人口の割合 x 人口密度"),
  custom.gof.names = c("R^2",
    "自由度決定済み R^2",
    "観測数"),
  custom.note = "注: 括弧内は標準偏差")
```

```
res <- list(model_2,model_4)
texreg(res,
  stars = c(0.001, 0.01, 0.05),
  digits = 3,
  caption = "説明変数が従属人口指数の逆数",
  caption.above = TRUE,
  custom.model.names = c("一人当たり所得 (一万円)", "県内総生産額 (百億円)"),
  custom.coef.name = c("従属人口指数の逆数",
    "人口密度 (千人/km^2)",
    "従属人口指数の逆数 x 人口密度"),
  custom.gof.names = c("R^2",
```

表 4: 説明変数が従属人口指数の逆数

	一人当たり所得 (一万円)	県内総生産額 (百億円)
従属人口指数の逆数	3106.864*** (623.175)	3875.173 (4388.189)
人口密度 (千人/km <sup>2</sup> )	79.400*** (4.997)	1518.417*** (35.187)
従属人口指数の逆数 x 人口密度	391.138*** (100.030)	-677.866 (704.375)
R <sup>2</sup>	0.207	0.680
自由度決定済み R <sup>2</sup>	0.205	0.679
観測数	1175	1175

注: 括弧内は標準偏差

"自由度決定済み R<sup>2</sup>",  
"観測数"),  
custom.note = "注: 括弧内は標準偏差")

- 応答変数 (GDP と一人当たり所得) の予測値を示す

model-1

$$\begin{aligned}\hat{Y}_{it} &= 0.014X_{it} + 0.505Z_{it} + 0.003(X_{it} \cdot Z_{it}) \\ &= 0.505Z_{it} + (0.014 + 0.003Z_{it})X_{it}\end{aligned}$$

model-2

$$\begin{aligned}\hat{Y}_{it} &= 31.069X_{it} + 0.794Z_{it} + 3.911(X_{it} \cdot Z_{it}) \\ &= 0.794Z_{it} + (31.069 + 3.911Z_{it})X_{it}\end{aligned}$$

model-3

$$\begin{aligned}\hat{Y}_{it} &= 0.102X_{it} + 15.729Z_{it} - 0.015(X_{it} \cdot Z_{it}) \\ &= 15.729Z_{it} + (0.102 - 0.015Z_{it})X_{it}\end{aligned}$$

model-4

$$\begin{aligned}\hat{Y}_{it} &= 38.752X_{it} + 15.184Z_{it} - 6.779(X_{it} \cdot Z_{it}) \\ &= 15.184Z_{it} + (38.752 - 6.779Z_{it})X_{it}\end{aligned}$$

調整変数が代表的な値を取るときの切片と傾きを図示する

#調整変数の 10 標準偏差  
sd(trans\$pop\_dens) \* 10

```
[1] 0.7229681
```

```
sd <- 0.7229681
```

```
0.505 * -sd
```

```
[1] -0.3650989
```

```
0.014 + 0.003 * -sd
```

```
[1] 0.0118311
```

表 5: model\_1 の切片と傾き

平均-10 標準偏差	平均	平均 +10 標準偏差
-0.365	0	\$0.36\$5
0.012	0.014	0.016

```
plot_t <- ggplot(trans, aes(x = pop_bon, y = income)) +  
  geom_point(pch = 16) +  
  geom_abline(intercept = 0.36, slope = 0.016,  
              linetype = "dashed") +  
  geom_abline(intercept = 0, slope = 0.014) +  
  geom_abline(intercept = -0.36, slope = 0.012,  
              linetype = "dotted")+  
  ylim(-1,1) +  
  labs(x = "労働者の割合 (%)", y = "一人当たり所得 (一万円)") +  
  geom_text(label = "(人口密度:平均 - 10 標準偏差)",  
            x = 2, y = -0.5, family = my_font) +  
  geom_text(label = "(人口密度:平均 + 10 標準偏差)",  
            x = 2, y = 0.5, family = my_font)+  
  geom_text(label = "(人口密度:平均)",  
            x = 2, y = -0.05, family = my_font)  
plot(plot_t)
```



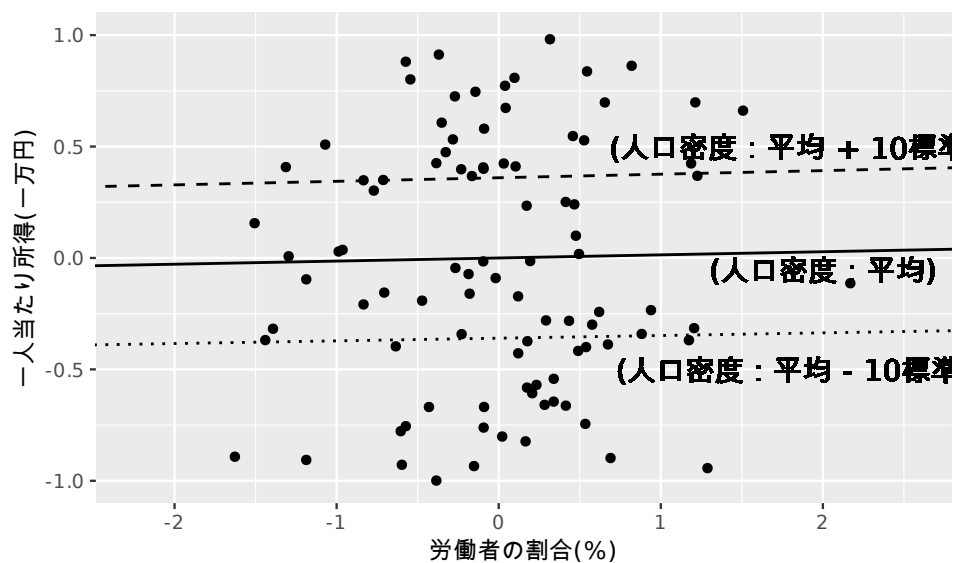


図 1: 複数の回帰直線を描いた

```
sd <- 0.7229681
```

```
0.794 * -sd
```

```
[1] -0.5740367
```

```
31.069 + 3.911 * -sd
```

```
[1] 28.24147
```

表 6: model\_2 の切片と傾き

平均-標準偏差	平均	平均 + 標準偏差
-0.574	0	0.574
28.241	31.069	33.896

```
plot_t <- ggplot(trans, aes(x = ration_rec, y = income)) +
  geom_point(pch = 16) +
  geom_abline(intercept = 0.574, slope = 33.896,
    linetype = "dashed") +
  geom_abline(intercept = 0, slope = 31.069) +
  geom_abline(intercept = -0.574, slope = 28.241,
    linetype = "dotted")+
  ylim(-1,1) +
  labs(x = "従属人口指数の逆数", y = "一人当たり所得 (一万円)") +
```

```
geom_text(label = "(人口密度:平均 - 10 標準偏差)",
          x = 0.002, y = -0.6, family = my_font) +
geom_text(label = "(人口密度:平均 + 10 標準偏差)",
          x = 0.002, y = 0.8, family = my_font)+
geom_text(label = "(人口密度:平均)",
          x = 0.002, y = -0.1, family = my_font)
plot(plot_t)
```

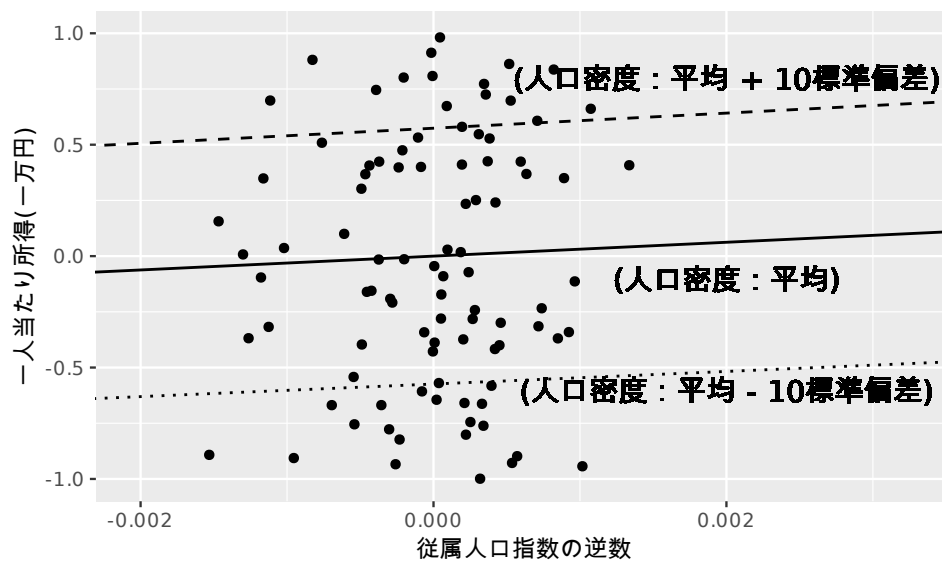


図 2: 複数の回帰直線を描いた

```
sd <- 0.7229681
```

```
15.729 * -sd
```

```
[1] -11.37157
```

```
0.102 - 0.015 * -sd
```

```
[1] 0.1128445
```

表 7: model\_3 の切片と傾き

平均-標準偏差	平均	平均 + 標準偏差
-11.372	0	11.371
0.113	0.102	0.091

```

plot_t <- ggplot(trans, aes(x = pop_bon, y = gdp)) +
  geom_point(pch = 16) +
  geom_abline(intercept = 11.371, slope = 0.091,
    linetype = "dashed") +
  geom_abline(intercept = 0, slope = 0.102) +
  geom_abline(intercept = -11.372, slope = 0.113,
    linetype = "dotted")+
  ylim(-20,20) +
  labs(x = "労働者の割合 (%)", y = "県内総生産額 (百億円)") +
  geom_text(label = "(人口密度:平均 - 10 標準偏差)",
    x = 2, y = -9, family = my_font) +
  geom_text(label = "(人口密度:平均 + 10 標準偏差)",
    x = 2, y = 10, family = my_font)+
  geom_text(label = "(人口密度:平均)",
    x = 2, y = -2, family = my_font)
plot(plot_t)

```

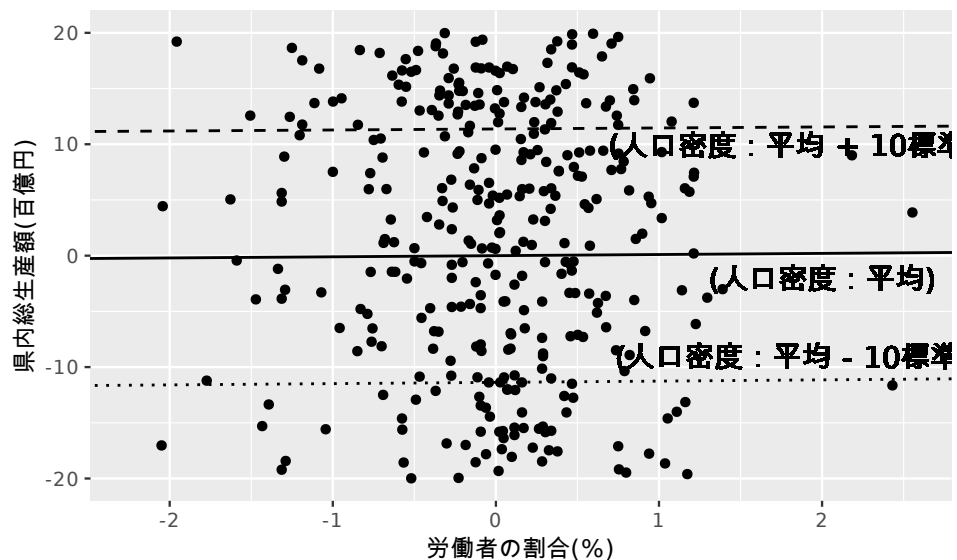


図 3: 複数の回帰直線を描いた

```
sd <- 0.7229681
```

```
15.184 * -sd
```

```
[1] -10.97755
```

```
38.752 - 6.779 * -sd
```

```
[1] 43.653
```

表 8: model\_4 の切片と傾き

平均-標準偏差	平均	平均 + 標準偏差
-10.978	0	10.978
43.653	38.752	33.851

```
plot_t <- ggplot(trans, aes(x = ration_rec, y = gdp)) +  
  geom_point(pch = 16) +  
  geom_abline(intercept = 10.978, slope = 33.851,  
             linetype = "dashed") +  
  geom_abline(intercept = 0, slope = 38.752) +  
  geom_abline(intercept = -10.978, slope = 43.653,  
             linetype = "dotted")+  
  ylim(-20,20) +  
  labs(x = "従属人口指数の逆数", y = "県内総生産額 (百億円)") +  
  geom_text(label = "(人口密度:平均 - 10 標準偏差)",  
           x = 0.0025, y = -12, family = my_font) +  
  geom_text(label = "(人口密度:平均 + 10 標準偏差)",  
           x = 0.0025, y = 9, family = my_font)+  
  geom_text(label = "(人口密度:平均)",  
           x = 0.0025, y = -2, family = my_font)  
plot(plot_t)
```

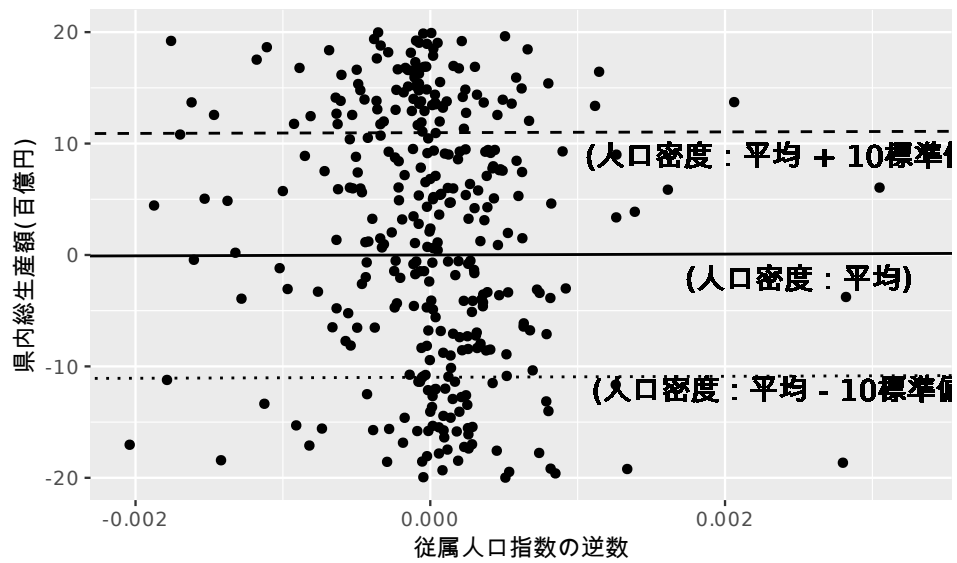


図 4: 複数の回帰直線を描いた

### 4.3 解釈

人口密度が高かろうが低かろうが, 若年層が経済に与える影響は大きくない. 人口密度が高いことの方が重要である.

## 5 test

- 限界効果

```
#model_1
beta_1 <- t(data.frame(rep(tidy(model_1)[1,2],100)))

beta_3 <- t(data.frame(rep(tidy(model_1)[3,2],100)))

values_X_2 <- data.frame(seq(min(trans$pop_dens), max(trans$pop_dens), length.out = 100))

df <- cbind(beta_1 ,beta_3,values_X_2)

df <- df|>
  rename(values_X_2 = seq.min.trans.pop_dens...max.trans.pop_dens...length.out...100.) |>
  mutate(marginal = beta_1 + beta_3 * values_X_2 )
```

```

values_var1 <- data.frame(seq(min(trans$pop_bon),
                             max(trans$pop_bon),
                             length.out = 100))
values_var2 <- data.frame(seq(min(trans$pop_dens),
                             max(trans$pop_dens),
                             length.out = 100))
values_var3 <- data.frame(seq(min(trans$pop_bon_X_pop_dens),
                             max(trans$pop_bon_X_pop_dens),
                             length.out = 100))

new_data <- cbind(values_var1, values_var2, values_var3)

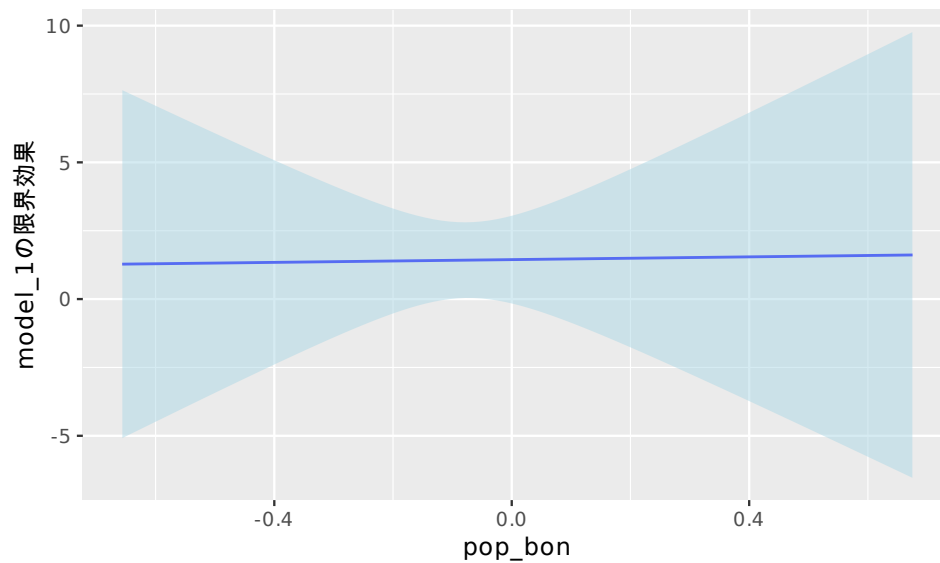
new_data <- new_data |>
  rename(pop_bon = seq.min.trans.pop_bon...max.trans.pop_bon...length.out...100.,
         pop_dens = seq.min.trans.pop_dens...max.trans.pop_dens...length.out...100.,
         pop_bon_X_pop_dens = seq.min.trans.pop_bon_X_pop_dens...max.trans.pop_bon_X_pop_dens...)

predictions <- predict(model_1, newdata = new_data, se.fit = TRUE)

critical_value <- (qnorm(0.05/ 2, lower.tail = FALSE))
df <- df |>
  mutate(upper_conf = marginal + critical_value * predictions$se.fit,
         lower_conf = marginal - critical_value * predictions$se.fit) |>
  cbind(values_var2) |>
  rename(pop_dens = seq.min.trans.pop_dens...max.trans.pop_dens...length.out...100.,)

#model_1
ggplot(df, aes(x = pop_dens, y = marginal)) +
  geom_line(color = "blue") +
  geom_ribbon(aes(ymin = lower_conf, ymax = upper_conf),
            fill = "lightblue", alpha = 0.5) +
  labs(x = "pop_bon", y = "model_1 の限界効果")

```



```
#model_2
beta_1 <- t(data.frame(rep(tidy(model_2)[1,2],100)))

beta_3 <- t(data.frame(rep(tidy(model_2)[3,2],100)))

values_X_2 <- data.frame(seq(min(trans$pop_dens), max(trans$pop_dens), length.out = 100))

df <- cbind(beta_1 ,beta_3,values_X_2)

df <- df|>
  rename(values_X_2 = seq.min.trans.pop_dens...max.trans.pop_dens...length.out...100.) |>
  mutate(marginal = beta_1 + beta_3 * values_X_2 )

values_var1 <- data.frame(seq(min(trans$ration_rec),
                             max(trans$ration_rec),
                             length.out = 100))
values_var2 <- data.frame(seq(min(trans$pop_dens),
                             max(trans$pop_dens),
                             length.out = 100))
values_var3 <- data.frame(seq(min(trans$ration_rec_X_pop_dens),
                             max(trans$ration_rec_X_pop_dens),
                             length.out = 100))
```

```

new_data <- cbind(values_var1,values_var2,values_var3)

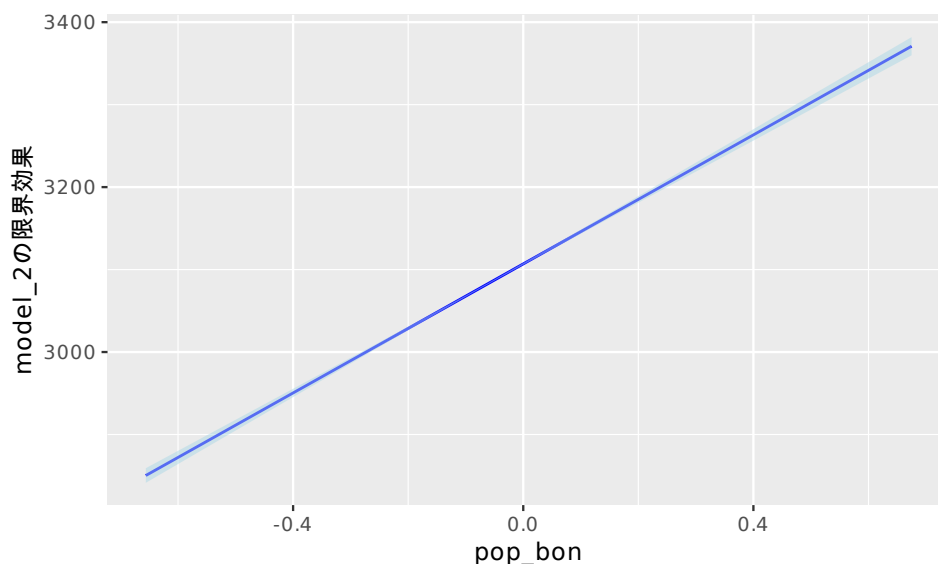
new_data <- new_data |>
  rename(ration_rec = seq.min.trans.ration_rec...max.trans.ration_rec...length.out...100.,
         pop_dens = seq.min.trans.pop_dens...max.trans.pop_dens...length.out...100.,
         ration_rec_X_pop_dens = seq.min.trans.ration_rec_X_pop_dens...max.trans.ration_rec_X_pop_de

predictions <- predict(model_2,newdata = new_data, se.fit = TRUE)

critical_value <- (qnorm(0.05/ 2, lower.tail = FALSE))
df <- df |>
  mutate(upper_conf = marginal + critical_value * predictions$se.fit,
         lower_conf = marginal - critical_value * predictions$se.fit) |>
  cbind(values_var2) |>
  rename(pop_dens = seq.min.trans.pop_dens...max.trans.pop_dens...length.out...100.,)

#model_2
ggplot(df, aes(x = pop_dens, y = marginal)) +
  geom_line(color = "blue") +
  geom_ribbon(aes(ymin = lower_conf, ymax = upper_conf),
            fill = "lightblue", alpha = 0.5) +
  labs(x = "pop_bon", y = "model_2 の限界効果")

```



```

#model_3
beta_1 <- t(data.frame(rep(tidy(model_3)[1,2],100)))

```



```

beta_3 <- t(data.frame(rep(tidy(model_3)[3,2],100)))

values_X_2 <- data.frame(seq(min(trans$pop_dens), max(trans$pop_dens), length.out = 100))

df <- cbind(beta_1 ,beta_3,values_X_2)

df <- df|>
  rename(values_X_2 = seq.min.trans.pop_dens...max.trans.pop_dens...length.out...100.) |>
  mutate(marginal = beta_1 + beta_3 * values_X_2 )

values_var1 <- data.frame(seq(min(trans$pop_bon),
                             max(trans$pop_bon),
                             length.out = 100))
values_var2 <- data.frame(seq(min(trans$pop_dens),
                             max(trans$pop_dens),
                             length.out = 100))
values_var3 <- data.frame(seq(min(trans$pop_bon_X_pop_dens),
                             max(trans$pop_bon_X_pop_dens),
                             length.out = 100))

new_data <- cbind(values_var1,values_var2,values_var3)

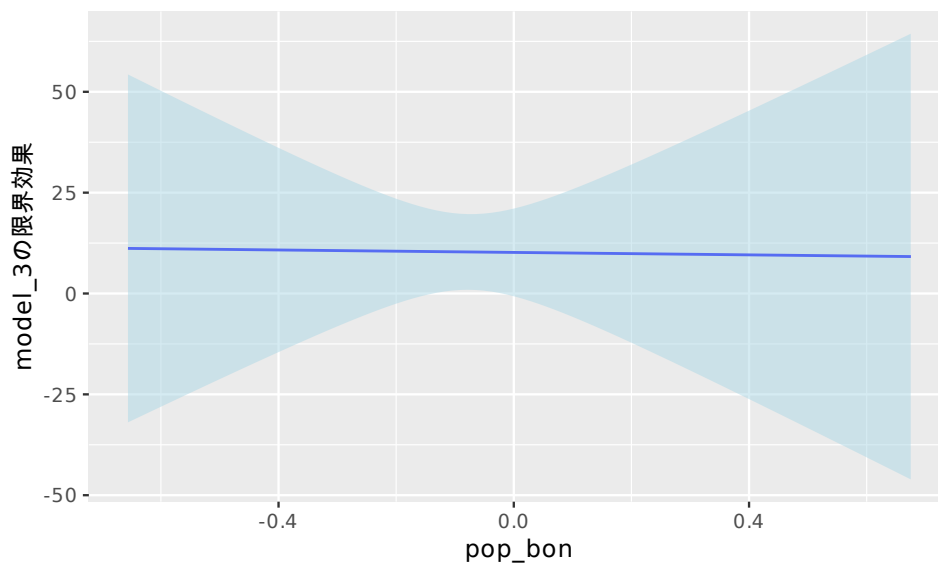
new_data <- new_data |>
  rename(pop_bon = seq.min.trans.pop_bon...max.trans.pop_bon...length.out...100.,
         pop_dens = seq.min.trans.pop_dens...max.trans.pop_dens...length.out...100.,
         pop_bon_X_pop_dens = seq.min.trans.pop_bon_X_pop_dens...max.trans.pop_bon_X_pop_dens...)

predictions <- predict(model_3,newdata = new_data, se.fit = TRUE)

critical_value <- (qnorm(0.05/ 2, lower.tail = FALSE))
df <- df |>
  mutate(upper_conf = marginal + critical_value * predictions$se.fit,
         lower_conf = marginal - critical_value * predictions$se.fit) |>
  cbind(values_var2) |>
  rename(pop_dens = seq.min.trans.pop_dens...max.trans.pop_dens...length.out...100.,)

```

```
#model_3
ggplot(df, aes(x = pop_dens, y = marginal)) +
  geom_line(color = "blue") +
  geom_ribbon(aes(ymin = lower_conf, ymax = upper_conf),
            fill = "lightblue", alpha = 0.5) +
  labs(x = "pop_bon", y = "model_3 の限界効果")
```



```
#model_4
beta_1 <- t(data.frame(rep(tidy(model_4)[1,2],100)))

beta_3 <- t(data.frame(rep(tidy(model_4)[3,2],100)))

values_X_2 <- data.frame(seq(min(trans$pop_dens), max(trans$pop_dens), length.out = 100))

df <- cbind(beta_1 ,beta_3,values_X_2)

df <- df|>
  rename(values_X_2 = seq.min.trans.pop_dens...max.trans.pop_dens...length.out...100.) |>
  mutate(marginal = beta_1 + beta_3 * values_X_2 )

values_var1 <- data.frame(seq(min(trans$ration_rec),
                             max(trans$ration_rec),
```

```

length.out = 100))
values_var2 <- data.frame(seq(min(trans$pop_dens),
                             max(trans$pop_dens),
                             length.out = 100))
values_var3 <- data.frame(seq(min(trans$ration_rec_X_pop_dens),
                             max(trans$ration_rec_X_pop_dens),
                             length.out = 100))

new_data <- cbind(values_var1, values_var2, values_var3)

new_data <- new_data |>
  rename(ration_rec = seq.min.trans.ration_rec...max.trans.ration_rec...length.out...100.,
         pop_dens = seq.min.trans.pop_dens...max.trans.pop_dens...length.out...100.,
         ration_rec_X_pop_dens = seq.min.trans.ration_rec_X_pop_dens...max.trans.ration_rec_X_pop_dens...length.out...100.)

predictions <- predict(model_4, newdata = new_data, se.fit = TRUE)

critical_value <- (qnorm(0.05/ 2, lower.tail = FALSE))
df <- df |>
  mutate(upper_conf = marginal + critical_value * predictions$se.fit,
         lower_conf = marginal - critical_value * predictions$se.fit) |>
  cbind(values_var2) |>
  rename(pop_dens = seq.min.trans.pop_dens...max.trans.pop_dens...length.out...100.,)

#model_4
ggplot(df, aes(x = pop_dens, y = marginal)) +
  geom_line(color = "blue") +
  geom_ribbon(aes(ymin = lower_conf, ymax = upper_conf),
            fill = "lightblue", alpha = 0.5) +
  labs(x = "pop_bon", y = "model_4 の限界効果")

```

