人口ー演習と課題

新保一成

2016 年 *5* 月 *17* 日版

Contents

準備

授業 Web サイトに記した「課題に向けた準備」のうち 1, 3, 4 を完了していること.

演習 1 — 世界総人口の推移

国連世界人口予測 (UN World Population Prosepect) を使って，1950 年から 2015 年までの世界人口の推移を横軸 (x 軸) に年次，縦軸 (y 軸) に世界総人口 (百万人単位) に取った折れ線グラフで表わせ.

データ

新保研 Web データのページから「男女別総人口」(ファイル番号 2) のデータファイル WPP2015_DB02_Populations_Annual.dsv を RStudio のプロジェクト・ディレクトリ (フォルダ) にダウンロードする．データファイルに収録されている変数の定義も同サイトにある.

データの読み込み

sqldf パッケージの **read.csv.sql** 関数で WPP2015_DB02_Populations_Annual.dsv を使用するデータを指定して読み込む.

```
library(sqldf)
popWorld <- read.csv.sql(" WPP2015_DB02_Populations_Annual.dsv" ,
  sep = " |" , eol = " \n" ,
  sql = " select LocID, Location, Time, PopTotal from file where VarID = 2 and LocID = 900" )
```

1. 1 行目で sqldf パッケージを呼び出す.
2. read.csv.sql の 1 番目の引数はファイル名で, 'file =' が省略されている.
3. データの区切り記号が縦棒であることを sep ="|" で与えている.
4. データファイルの改行コードがラインフィード (LF) であることを eol = "\n" で指示. これは, Windows の場合のみ必要.
5. sql = に続く文が **SQL の SELECT 文**でファイルの中からどのデータをどのような条件で抽出するかを指示する.
    a. この SELECT 文では国・地域コード (LocID), 国・地域名 (Location), 年次 (Time), 総人口 (PopTotal, 千人単位) を取り込むことを指示.
    b. 'from file' で'file =' で指定したテーブルから読むことを指示.
    c. '**where**' 以下が抽出条件で, 中位推計を使うことを'VarID = 2' で, 世界全体だけを取り込むことを'LocID = 900' で指示. 2 つの条件が同時に成り立つことを指示するために'and' で条件を結合.

読み込んだデータを確認する

- read.csv.sql で読み込まれたデータは, R のデータフレーム・オブジェクトになる.
- データフレーム popWorld は, LocID(国・地域コード), Location(国・地域名), Time(年次), PopTotal(総人口) で構成される.
- RStudio の右上の [Environment] の [Data] にリストされている popTotal をクリックすると, 左上の Data Viewer にその内容が表示される.
- あるいは左下の [Console] から以下を入力. **head** はデータの頭の部分を出力する関数である. 間違いなく世界総人口の時系列が取り込まれたことを確認しよう.

```
head(popWorld)
```

```
##   LocID Location Time PopTotal
## 1   900    World 1950  2525149
## 2   900    World 1951  2571868
## 3   900    World 1952  2617940
## 4   900    World 1953  2664029
## 5   900    World 1954  2710678
## 6   900    World 1955  2758315
```

dplyr でデータを整理する

問題にあるグラフを書くためには, Time と PopTotal の 2 つの変数があれば十分である. ここで, dplyr の **select** 関数を使って, データフレーム popWorld に Time と PopTotal だけを残してみよう.

```
library(dplyr)
popWorld <- select(popWorld, Time, PopTotal)
head(popWorld)
```

```
##   Time PopTotal
## 1 1950  2525149
## 2 1951  2571868
## 3 1952  2617940
## 4 1953  2664029
## 5 1954  2710678
## 6 1955  2758315
```

dplyr と ggplot2 でグラフを描く

```
library(ggplot2)
popWorldLinePlot <- popWorld %>%
  filter(Time <= 2015) %>%
  ggplot(aes(x = Time, y = PopTotal / 1000)) + geom_line() +
  xlab(" Year " ) + ylab(" Total Population in the World(Million)" )
```

dplyr とデータパイプライン

dplyr とデータパイプラインを上手く使うと効率よくプログラムすることができる.

- 最初の 2 行 **dplyr** パッケージと **ggplot2** パッケージを呼び出す.

%>% がデータパイプラインの役目を果たしていて，'popWorldLinePlot <- popWorld %>%' 以下の部分は次のように書くのと同じである.

```
popWorld2015 <- filter(popWorld, Time <= 2015)
popTotalLinePlot <- ggplot(popTotal2015, aes...)
```

- '%>%' は dplyr が提供するパイプ機能で，'%>%' の左側で処理されたデータがその右側に自動的に渡される．パイプ機能を上手く使うと popWorld2015 のような中間なデータを生成しないで済む.
- dplyr の **filter** 関数は，渡されたデータを条件にしたがってフィルタリングする関数である．popWorld から 2015 年以前のデータのみを抽出することを'Time <= 2015' で指示している.
- 'popWorld %>% filter(Time <= 2015) %>% ggplot(...)' で popWorld がパイプを通じて filter に渡され，さらに次のパイプで濾過されたデータが ggplot に渡される.

ggplot2 で折れ線グラフを描く

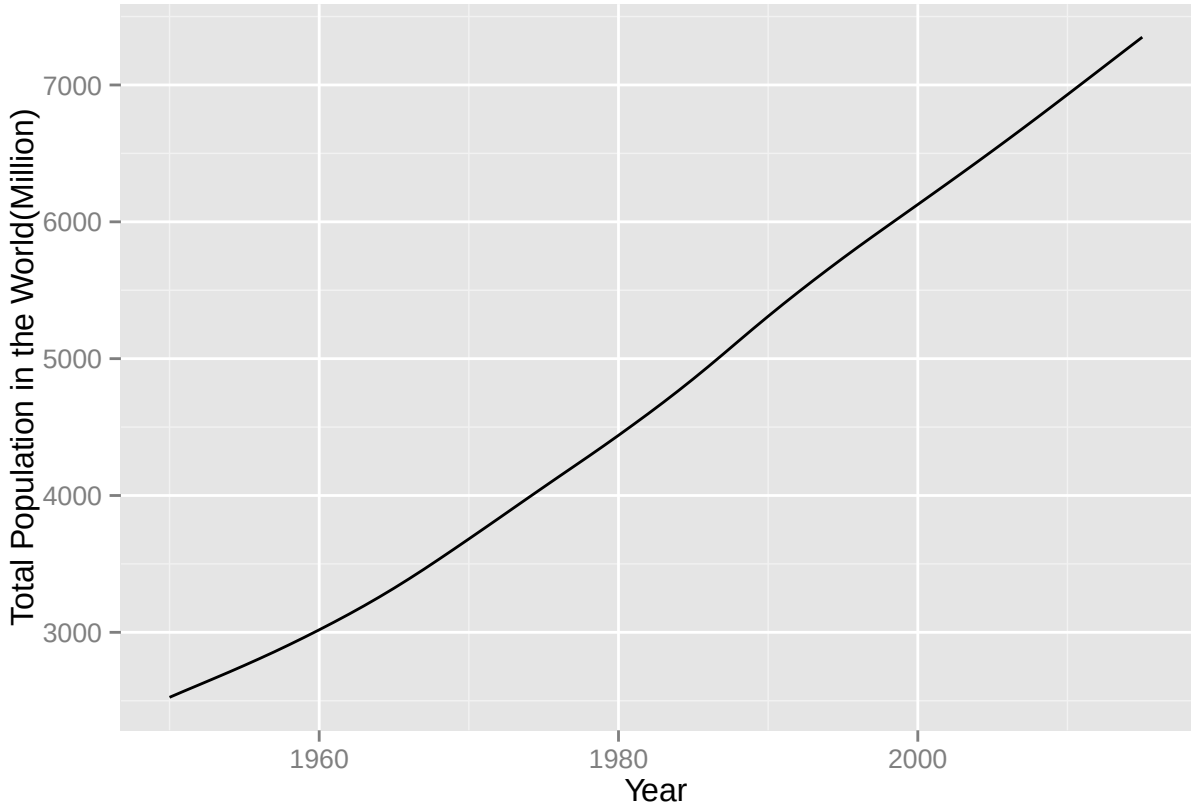
ggplot2 は，データを与える，グラフの種類の選択，ラベル，凡例，軸の設定などを'+' 記号でつなぎながら階層的にデータを視覚化する.

- **aes** 関数で x 軸と y 軸のデータを指示する．'y = PopTotal / 1000' で人口の単位を千人から百万人に変換.
- aes で指示したデータを折れ線グラフにすることを'**geom_line()**' で指示.
- **xlab** と **ylab** で x 軸と y 軸のラベルを指示.
- デフォルトでは軸テキストのサイズが小さく，プレゼンテーション等で見難い時もある．軸テキストのサイズを変更するには，'**+ theme(text = element_text(size = 20))**' とする。

グラフの出力

ggplot2 で生成されたグラフは，print 関数で出力することができる．RStudio 左下の [Console] から'print(popWorldLinePlot)' と入力すると右下のウインドウに下のグラフが表示される.

```
print(popWorldLinePlot)
```



ggsave 関数を使うと，その直前に作成されたグラフを PDF ファイルとして保存することができる．下の例で，"plotWorldLinePlot.pdf" は PDF ファイル名で，保存先は現在のワーキング・ディレクトリになる.

```
ggsave(" plotWorldLinePlot.pdf" )
```

R スクリプトのまとめ

```
library(sqldf)
library(dplyr)
library(ggplot2)
popWorld <- read.csv.sql(" WPP2015_DB02_Populations_Annual.dsv" ,
  sep = " |" , eol = " \n" ,
  sql = " select LocID, Location, Time, PopTotal from file where VarID = 2 and LocID = 900" )
popWorldLinePlot <- popWorld %>%
  filter(Time <= 2015) %>%
  ggplot(aes(x = Time, y = PopTotal / 1000)) + geom_line() +
  xlab(" Year" ) + ylab(" Total Population in the World(Million)" ) +
  theme(text = element_text(size = 20))
  ggsave(" plotWorldLinePlot.pdf" )
```

演習 2 — 世界総人口の推移を先進国と開発途上国に分ける

演習 1 と同じデータセットを使って，1950 年から 2015 年までの世界人口の推移を横軸 (x 軸) に年次，縦軸 (y 軸) に世界総人口 (百万人単位) に取った先進国と開発途上国の積み上げ面グラフ (stacked area graph) で表わせ．続いて，世界総人口にしめる開発途上国のウエイトがわかりやすいように，開発途上国と先進国のシェアで測った積み上げ面グラフを描け．

データの読み込みと確認

演習 1 と同じデータセットから，国連が定義する More developed regions(LocID = 901) を先進国，Less developed regions(LocID = 902) を開発途上国として，この 2 地域のデータを読み込むことにする．国・地域コード (LocID) が 901 または 902 であるから，'or' を使って'LodID = 901 or LocID = 902' という具合に 2 つの条件をで結んで where 以下に与えればよい．抽出したい国・地域が数多くある場合に'or' で結合するのは煩雑なので，そのような場合には'**in**'を使うのが便利である．いまのケースでは，'LocID in(901, 902)'，と書く．

read.csv.sql 関数で WPP2015_DB02_Populations_Annual.dsv から先進国と開発途上国の総人口を読み込むには次のようにすればよい．

```
popRegion <- read.csv.sql(" WPP2015_DB02_Populations_Annual.dsv" ,
  sep = " |" , eol = " \n" ,
  sql = " select LocID, Location, Time, PopTotal from file where VarID = 2 and LocID in(901, 902)" )
```

head 関数で中味を確認する．

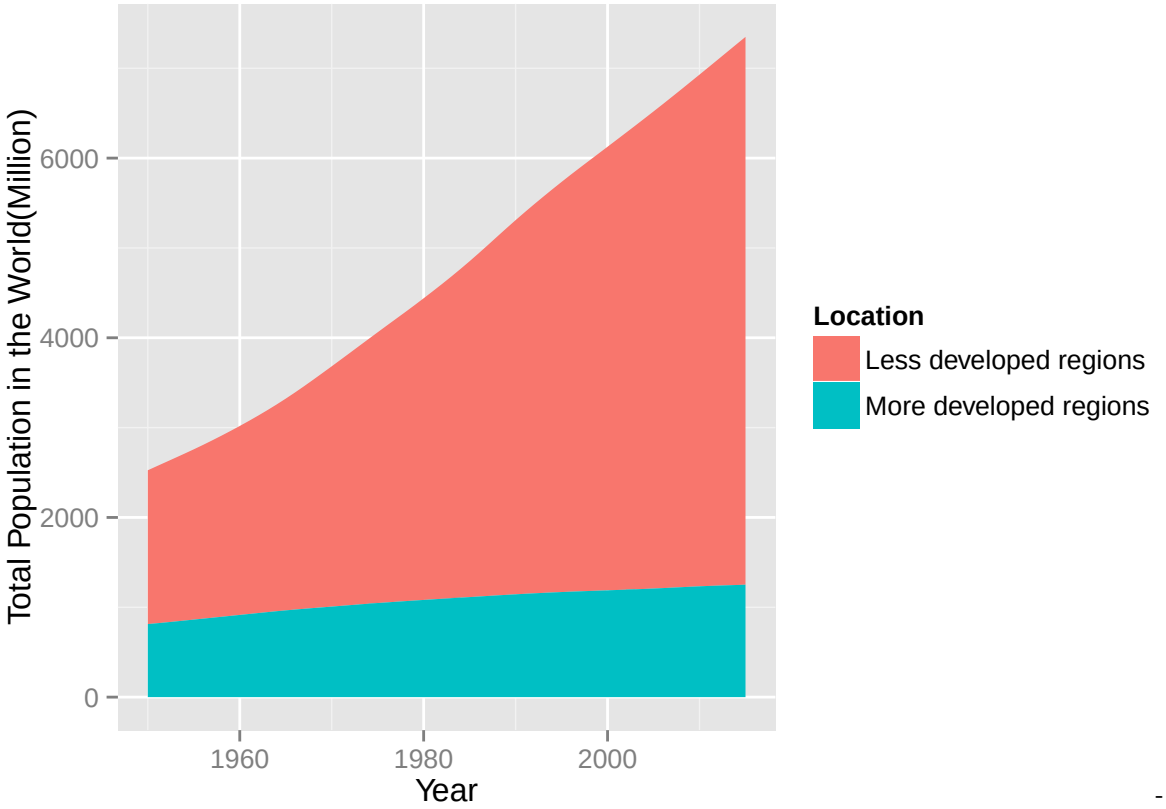
```
head(popRegion)
```

```
##   LocID              Location Time PopTotal
```

```
## 1   901 More developed regions 1950 812988.8
## 2   901 More developed regions 1951 822320.5
## 3   901 More developed regions 1952 832148.7
## 4   901 More developed regions 1953 842293.6
## 5   901 More developed regions 1954 852613.4
## 6   901 More developed regions 1955 863004.1
```

積み重ね領域グラフを描いて出力する

```
popRegionAreaPlot <- popRegion %>%
  filter(Time <= 2015) %>%
  ggplot(aes(x = Time, y = PopTotal / 1000, fill = Location)) + geom_area() +
  xlab(" Year " ) + ylab(" Total Population in the World(Million)" )
print(popRegionAreaPlot)
```

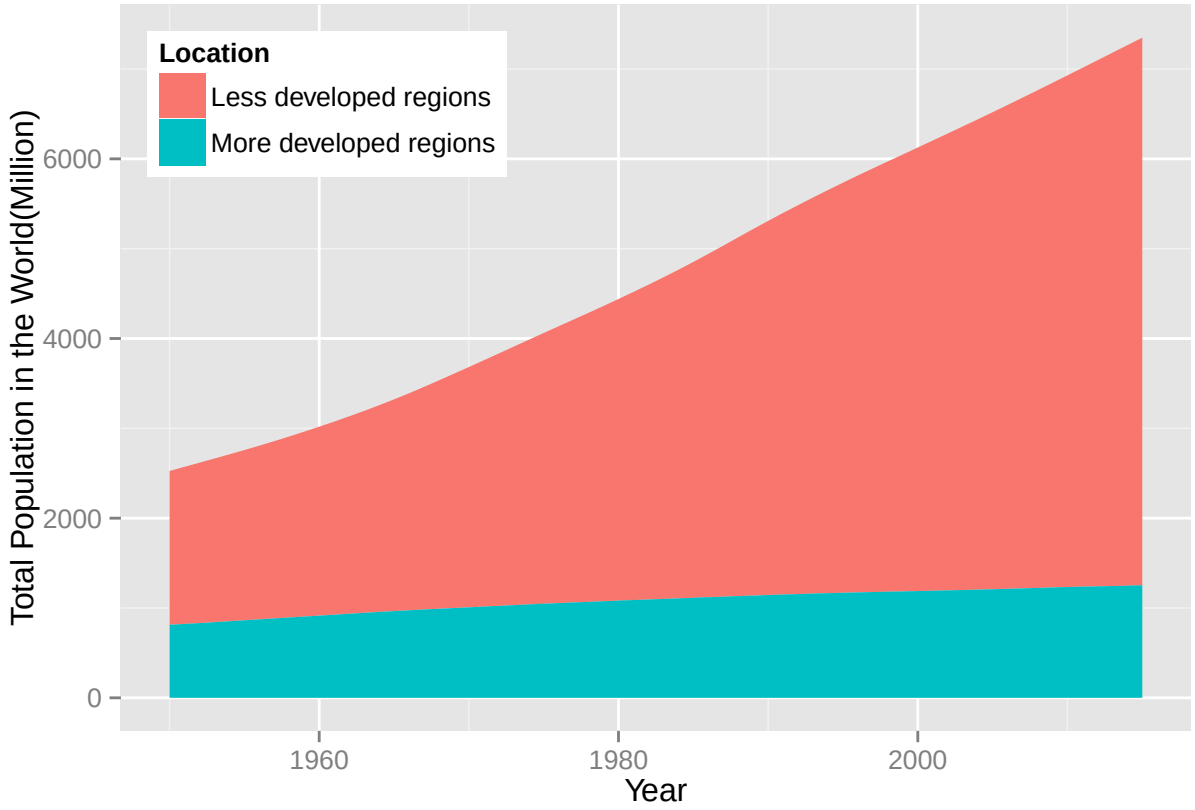


- グラフの形式を **geom_area** とすると aes に fill でマッピングした変数で積み上げ面グラフを描くことができる．- aes の'fill = Location' は，国・地域名で面をマッピングして塗りつぶすことを指示している．- デフォルトでは，グラフの右側に凡例が表示される．

凡例の位置を変えてみる

凡例をグラフの空間内に移動させて，グラフを大きく見せてみよう．

```
popRegionAreaPlot <- popRegionAreaPlot +
  theme(legend.position=c(0, 1), legend.justification=c(0, 1))
print(popRegionAreaPlot)
```



ここでおもしろいのは，出来上がったグラフ popTotalAreaPlot に凡例の位置を変える要素を加えることで変更を実現できることだ．凡例の位置を変更するには theme に **legend.position** と **legend.adjustment** で位置パラメタを与える．legend.position では左下が (0, 0) のポジションで，'legend.position = c(0, 1)' で凡例の位置を左上に変更することを指示している．legend.adjustment では凡例のどの位置を legend.position で指定した位置に持ってくるのかを指示する．デフォルは凡例の中心 (0.5, 0.5) である．したがって，'legend.adjustment = c(0, 1)' で凡例の左上がグラフの左上に位置するように調整している．

開発途上国と先進国の世界総人口にしめるシェアの計算

演習 1 で用いた世界総人口 (p) は，開発途上国人口 (p_{LD}) と先進国人口 (p_{MD}) の合計である．

$$p = p_{LD} + p_{MD}$$

各地域の世界総人口にしめるシェアは次のように計算される．

$$s_{LD} = \frac{p_{LD}}{p}, \quad s_{MD} = \frac{p_{MD}}{p}, \quad s_{LD} + s_{MD} = 1$$

ここで s_{LD} と s_{MD} は，それぞれ開発途上国と先進国の世界総人口にしめるシェアであり，その合計は定義より 1 になる．

popRegion の 2000 年のデータを確認してみると次のようになっている．

```
filter(popRegion, Time == 2000)
```

```
##   LocID               Location Time PopTotal
## 1   901 More developed regions 2000  1188812
## 2   902 Less developed regions 2000  4937810
```

この両方の行に popWorld に含まれている 2000 年の世界総人口のデータ

```
filter(popWorld, Time == 2000)
```

```
##   Time PopTotal
## 1 2000  6126622
```

があれば，シェアの計算は簡単である．

dplyr が備えている **join** 関数は，2 つのデータフレームを両者に共通な変数をキーにしてマージする．特に，**left_join** は，left_join(x, y) としたときに，左側の x の全ての行と対応した y の行を含む．演習 1 で使用したデータフレーム popWorld と演習 2 の popRegion を年次 (Time) をキーにしてマージして，popRegion に世界総人口を取り込もう．まず，popWorld の世界総人口 PopTotal を PopWorld と名前を変更する．そうしないと，Time と PopTotal の 2 つの変数がキーになってしまうからである．

```
popWorld <- rename(popWorld, PopWorld = PopTotal)
popRegion <- left_join(popRegion, popWorld)
```

```
## Joining by: ” Time”
```

```
filter(popRegion, Time == 2000)
```

```
##   LocID               Location Time PopTotal PopWorld
## 1   901 More developed regions 2000  1188812  6126622
## 2   902 Less developed regions 2000  4937810  6126622
```

このように同じ年の 2 つの異なる地域に同じ世界総人口 PopWorld が対応したので，次のように dplyr の **mutate** 関数を使って各地域の人口シェアを計算することができる．mutate は，既存のデータフレームに新たな変数を定義する関数である．

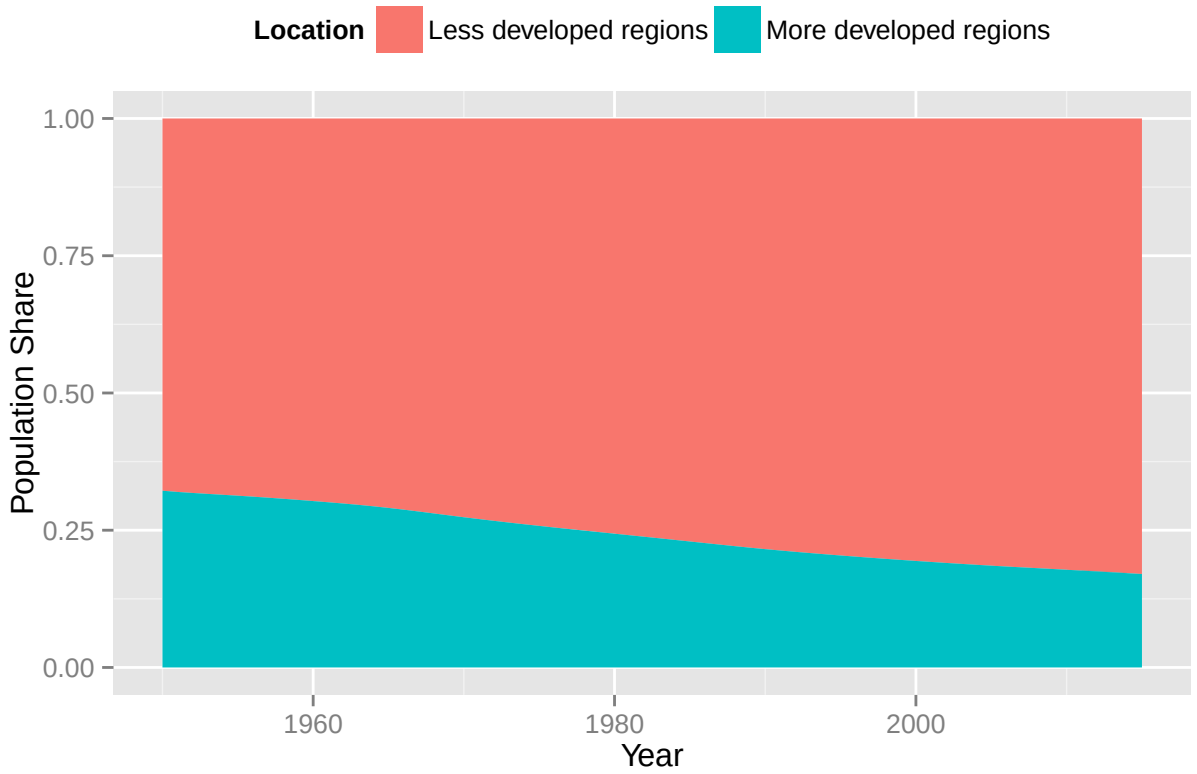
```
popRegion <- mutate(popRegion, PopShare = PopTotal / PopWorld)
filter(popRegion, Time == 2000)
```

```
##   LocID               Location Time PopTotal PopWorld  PopShare
## 1   901 More developed regions 2000  1188812  6126622 0.1940403
## 2   902 Less developed regions 2000  4937810  6126622 0.8059597
```

開発途上国と先進国のシェアの和が 1 になっていることも確認できる.

それではシェアの推移を積み上げ面グラフで描いてみよう.

```
popShareAreaPlot <- popRegion %>%
  filter(Time <= 2015) %>%
  ggplot(aes(x = Time, y = PopShare, fill = Location)) + geom_area() +
  xlab(" Year " ) + ylab(" Population Share " )
popShareAreaPlot <- popShareAreaPlot + theme(legend.position=" top " )
print(popShareAreaPlot)
```



ここでは, theme(legend.position="top") を追加して凡例の位置をグラフの上に変更した.