Rにおけるtidyなデータ処理

石田基広

2019年09月03日

自己紹介

第一部

コード

- Console
- Script
- Markdown

コードで操作するとは

すべて名前を指定して実行する

- オブジェクト
  - df：データの名前
- データとの紐付け:<-
  - df <- read.csv(“file.csv”)

コードで操作するとは

データを関数で操作

- 関数
  - head()：処理の名前
- head(df)：データの冒頭表示

データ

```
library(RMeCab)
df <-docDF("doc", type = 1)
```

```
head(df)
```

```
  TERM   POS1   POS2  1  2  3
1                 2     5     3
2                 2     2     2
3                0     0     2
4               0     1     0
5               1     1     0
6             *     0     1     0
```

データの構造

- ベクトル
  - c(1, 3, 5, 7, 9)
- データフレーム

TERM	POS	FREQ
企業	名詞	2
伝える	動詞	3
高い	形容詞	4

-

分析でよくある操作

特定の列（変数）の指定：$

```
#
df$TERM
```

```
 [1] " "        " "        " "      " "      " "      " "
 [7] " "        " "        " "      " "      " "      " "
[13] "  "    " "        "   " " "        " "      "  "
[19] "  "    " "
```

条件抽出

添字([行の指定 , 列の指定])を駆使

```
df [ df$TERM==" " , ]
```

```
   TERM POS1 POS2  1  2  3
81           1     0     2
```

条件抽出

各テキストでの出現回数を合算すると5を超える単語

```
df [ rowSums( df [,
    c(" 1"," 2"," 3")]) > 5, ]
```

```
   TERM POS1   POS2  1  2  3
1               2     5     3
2               2     2     2
31             3     5     1
32             2     4     2
39             2     3     5
91               1     5     2
```

列の追加

各テキストでの出現回数を合算した列

```
df$  <- rowSums(
  df[, c(" 1", " 2", " 3") ] )
```

```
  TERM  1  2  3
1         2     5     3   10
2         2     2     2    6
3       0     0     2    2
4       0     1     0    1
5       1     1     0    2
6         0     1     0    1
```

変数の加工

合計頻度列を標準化

```
df$   <- scale(df$ )
```

```
  TERM  1  2  3
1         2     5     3  4.14854297
2         2     2     2  2.08459124
3       0     0     2  0.02063952
4       0     1     0 -0.49534841
5       1     1     0  0.02063952
6         0     1     0 -0.49534841
```

tidyなデータ処理

tidy data

Hadley Wickham

- Each variable forms a column.
- Each observation forms a row.
- Each type of observational unit forms a table.

tidyverse

```
install.packages("tidyverse")
library(tidyverse)
```

tidy data

messy data

```
  TERM   POS1  1  2  3
1             2     5     3
2             2     2     2
3           0     0     2
4           0     1     0
5           1     1     0
6           0     1     0
```

tidy data

```
# A tibble: 300 x 4
   TERM  POS1   Doc    FREQ
   <chr> <chr>  <chr> <int>
 1           1     2
 2           1     2
 3         1     0
 4         1     0
 5         1     1
 6         1     0
 7           1     0
 8           1     0
 9         1     0
10         1     1
# ... with 290 more rows
```

tibble

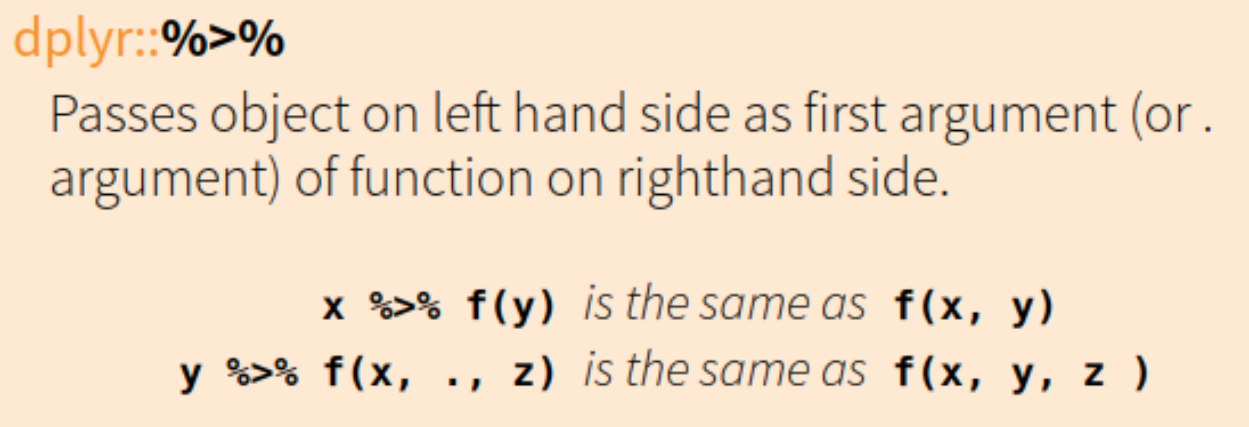
data.frameの拡張

```
tb <- tibble(
  Name = LETTERS[1:5],
  X = 1:5,
  Y = X^2
)
```

```
# A tibble: 5 x 3
  Name      X     Y
  <chr> <int> <dbl>
1 A         1     1
2 B         2     4
3 C         3     9
4 D         4    16
5 E         5    25
```

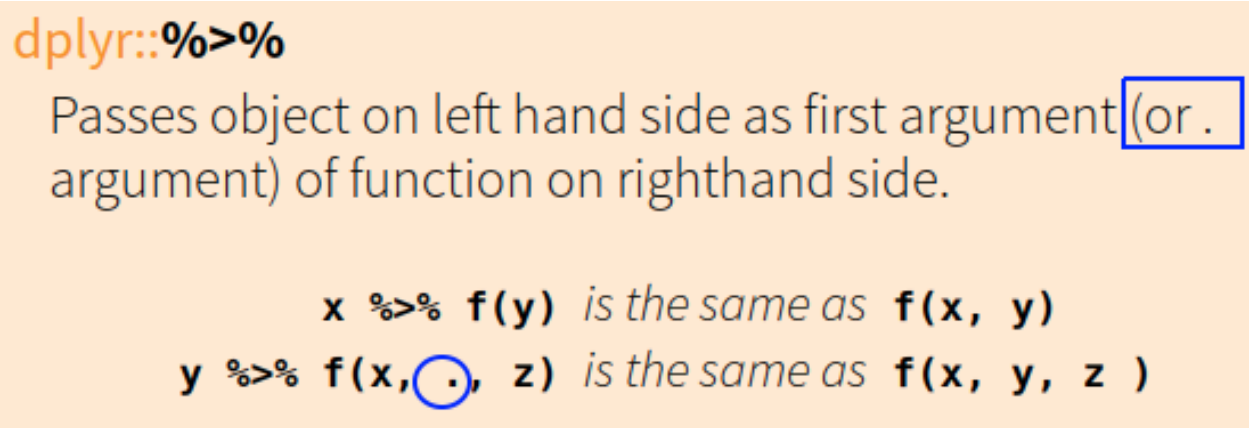
パイプ演算子

%>%



https://www.rstudio.com/wp-content/uploads/2015/02/data-wrangling-cheatsheet.pdf

ドット



Why Pipe

Name列がBのレコードのYの最大値

```
#
temp <- tb [ tb$Name=="B" , ]
max(temp $ Y)
```

```
[1] 4
```

Why Pipe

一時ファイルを作成しない方法

```
#
max( tb [ tb$Name=="B", ] $ Y )
```

```
[1] 4
```

Use Pipe

tidyverse流： filterとselectで抽出

```
# max(df2[df2$Name=="B",]$m)
tb %>% filter(Name == "B") %>%
               select(Y) %>% max()
```

```
[1] 4
```

列選択

select(列)

```
# tidyverse
tb %>% select(Name, Y) %>%
         arrange(desc(Name))
```

```
# A tibble: 5 x 2
  Name      Y
  <chr> <dbl>
1 E        25
2 D        16
3 C         9
4 B         4
5 A         1
```

列選択

- R本来の書き方
  - tb $ Y : (ベクトル)
- dplyr流
  - tb %>% select(Y) : (データフレーム)

返り値はデータフレーム

```
mean(tb$X)
```

```
[1] 3
```

```
tb %>% select(X) %>% mean()
```

```
[1] NA
```

```
tb %>% select(X) %>% pull() %>% mean()
```

```
[1] 3
```

条件抽出

```
# tidyverse
tb %>% filter(Y >= 5)
```

```
# A tibble: 3 x 3
  Name      X     Y
  <chr> <int> <dbl>
1 C         3     9
2 D         4    16
3 E         5    25
```

条件抽出

- R本来
  - tb [tb $ Y >= 5,]
- dplyr
  - tb %>% filter(Y >= 5)

変数変換

```
#
temp <- tb
temp $ Y <- log(temp$Y)
```

変数変換

```
# tb2 <- tb
# tidyverse
tb %>% mutate(Y = log(Y))
```

```
# A tibble: 5 x 3
  Name      X     Y
  <chr> <int> <dbl>
1 A         1  0
2 B         2  1.39
3 C         3  2.20
4 D         4  2.77
5 E         5  3.22
```

条件付き変換

```
dat
```

```
  Y X1 X2
1 A  2  3
```

```
2 A  3  4
3 B  4  5
4 B  5  6
5 C  6  7
6 C  7  8
```

条件付き変換

数値列だけ対数化

```
dat %>% mutate_if(is.numeric, log )
```

```
  Y        X1       X2
1 A 0.6931472 1.098612
2 A 1.0986123 1.386294
3 B 1.3862944 1.609438
4 B 1.6094379 1.791759
5 C 1.7917595 1.945910
6 C 1.9459101 2.079442
```

要約

```
dat %>% summarise(X1_mean = mean(X1),
                  X1_sd = sd(X1))
```

```
  X1_mean    X1_sd
1     4.5 1.870829
```

要約

数値列だけ平均を求める

```
dat %>% summarise_if(is.numeric, mean)
```

```
   X1  X2
1 4.5 5.5
```

要約

平均と分散

```
dat %>% summarise_if(is.numeric,
                     list(mean, var))
```

```
  X1_fn1 X2_fn1 X1_fn2 X2_fn2
1    4.5    5.5    3.5    3.5
```

要約

出力の列名を調整

```
dat %>% summarise_if(is.numeric,
          list( ~mean(.), ~sd(.)))
```

```
  X1_mean X2_mean    X1_sd    X2_sd
1     4.5     5.5 1.870829 1.870829
```

横型データ

```
yoko
```

```
# A tibble: 8 x 4
  Name  Time1 Time2 Time3
  <chr> <dbl> <dbl> <dbl>
1 A      0.1   0.18  0.11
2 B      0.3   0.33  0.35
3 C      0.2   0.22  0.26
4 D      0.44  0.47  0.43
5 E      0.51  0.56  0.55
6 F      0.6   0.66  0.68
7 G      0.77  0.78  0.72
8 H      0.81  0.88  0.86
```

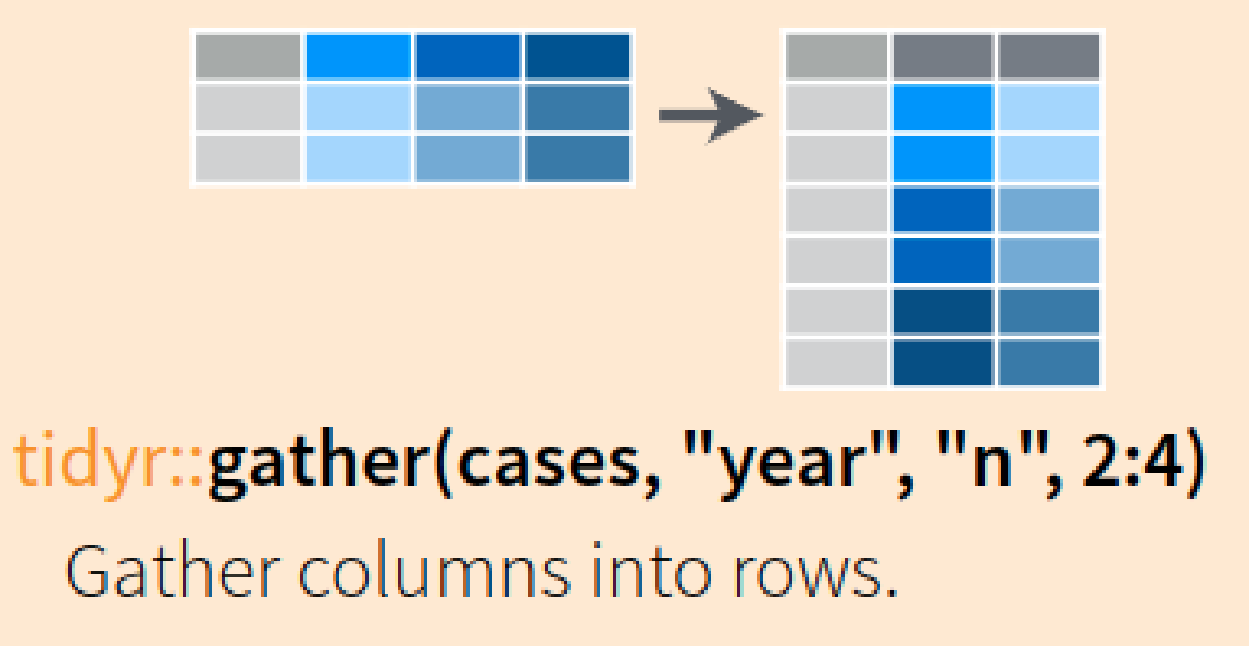
縦型データへ

```
yoko %>% gather(
  key = time,
  value = score,
  -Name)
```

```
# A tibble: 24 x 3
   Name  time  score
   <chr> <chr> <dbl>
 1 A     Time1  0.1
 2 B     Time1  0.3
 3 C     Time1  0.2
 4 D     Time1  0.44
 5 E     Time1  0.51
 6 F     Time1  0.6
 7 G     Time1  0.77
 8 H     Time1  0.81
 9 A     Time2  0.18
10 B     Time2  0.33
# ... with 14 more rows
```

gather

```
df %>% gather(key = year, value = n)
```



文書データを縦型へ

```
  TERM   POS1   POS2  1  2  3
1                 2     5     3
2                 2     2     2
3               0     0     2
4             0     1     0
5             1     1     0
6           *     0     1     0
```

文書データを縦型へ

```
library(tidyverse)
tidy_df <- df %>% gather(
          key = Doc, value = FREQ,
           1,  2,  3)
```

変換結果

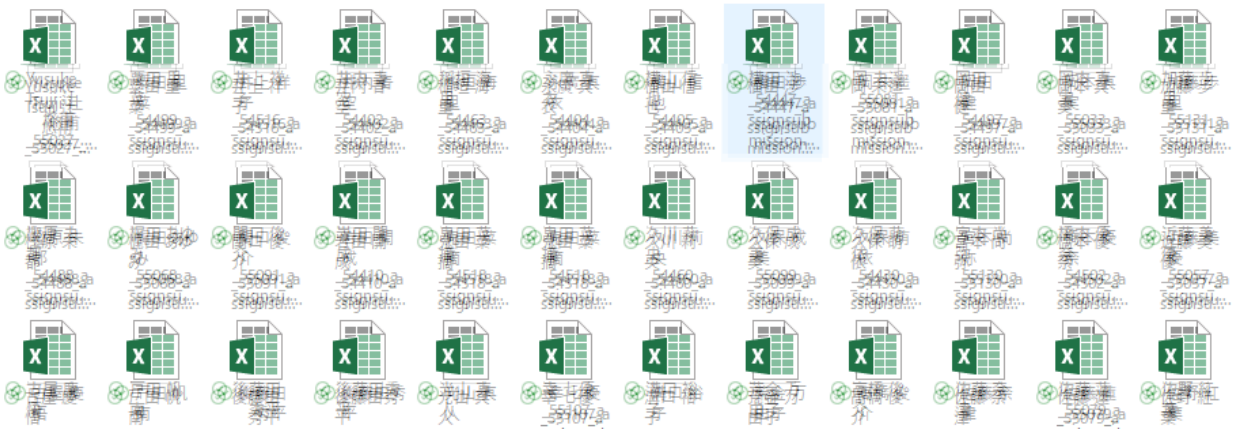
```
tidy_df %>% filter(TERM == " ")
```

```
  TERM POS1 POS2   Doc FREQ
1          1    2
2          2    2
3          3    2
```

実践

退屈なことはRにやらよう

たくさんのExcelファイルを採点



Excelファイル採点

	A	B
1		Heiaht
2		176.3
3		166.9
4		182.3
5		181.4
6		179.6
7		169.8
8		175.3
9		178.2
10		172.8
11	平均値	
12	標準偏差	
13		

・ 学生は指定したシートの指定したセルに式を入力している

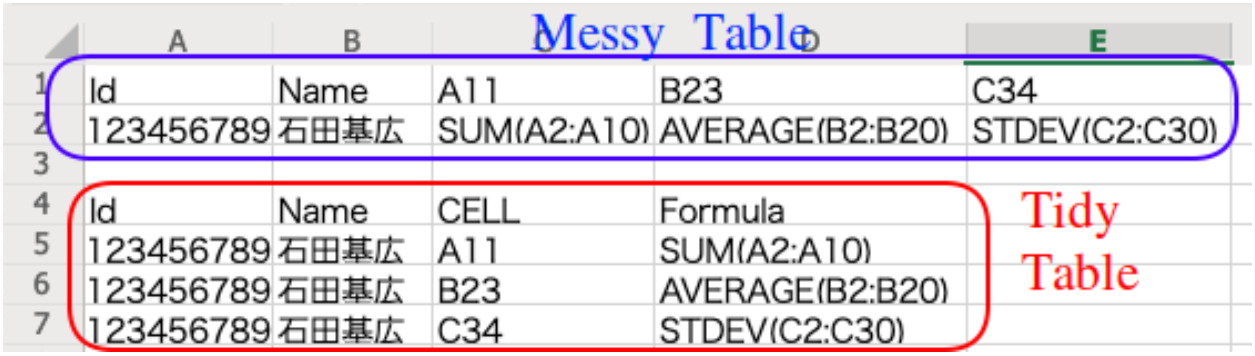
ファイル一覧

```
files <- list.files("C:\\kadai",
                    pattern = "xlsx")
```

ファイル一覧

```
> files %>% str_extract("\\d{8,10}\\p{Han}*.xlsx$")
  [1] "1017050[redacted].xlsx"          "１０１８０５[redacted]石田里菜.xlsx"
  [3] "101805[redacted]洋子.xlsx"       "101805153[redacted].xlsx"
  [5] "101805[redacted]海里.xlsx"       "1018051164[redacted].xlsx"
  [7] "101805[redacted]也.xlsx"         "1018051521[redacted]lsx"
  [9] "101805[redacted].xlsx"           "101805125[redacted]sx"
 [11] "101805[redacted]実.xlsx"         "101805006[redacted]xlsx"
 [13] "101805[redacted].xlsx"           "101805111[redacted]xlsx"
 [15] "101805[redacted]香.xlsx"         "101805052[redacted]xlsx"
 [17] "101805[redacted]太.xlsx"         NA
 [19] "101805[redacted]都.xlsx"         NA
```

採点用原簿



	A	B	C	D	E
1	Id	Name	A11	B23	C34
2	123456789	石田基広	SUM(A2:A10)	AVERAGE(B2:B20)	STDEV(C2:C30)
3					
4	Id	Name	CELL	Formula	
5	123456789	石田基広	A11	SUM(A2:A10)	
6	123456789	石田基広	B23	AVERAGE(B2:B20)	
7	123456789	石田基広	C34	STDEV(C2:C30)	

原簿と結合

作業ファイルを、学生番号で原簿ファイルと結合

	A	B	C	D	E	F	G	H
1	No	年度	所属名	学生番号	氏名	性別	確定した評価	
2	1	2018	地球科学	7019005101	[illegible]実	女		
3	2	2018	地球科学	7019005301	[illegible]未	女		
4	3	2018	地球科学	7019005401	[illegible]さと	女		
5	4	2018	地球科学	7019005501	[illegible]	女		
6	5	2018	地球科学	7019005601	[illegible]里	女		
7	6	2018	地球科学	7019005701	[illegible]	男		
8	7	2018	地球科学	7019005801	[illegible]太	男		
9	8	2018	地球科学	7019005901	[illegible]	女		
10	9	2018	地球科学	7019006101	[illegible]華	女		
11	10	2018	地球科学	7019006201	[illegible]法	男		
12	11	2018	地球科学	7019006401	[illegible]優	女		
13	12	2018	地球科学	7019006501	[illegible]尋	女		
14	13	2018	地球科学	7019006701	[illegible]里	女		
15	14	2018	地球科学	7019006801	[illegible]	男		
16	15	2018	地球科学	7019006901	[illegible]菜	女		
17	16	2018	地球科学	7019007001	[illegible]咲	女		
18	17	2018	地球科学	7019007201	[illegible]衣	女		
19	18	2018	地球科学	7019007301	[illegible]	男		
20	19	2018	地球科学	7019007401	[illegible]弥	男		

リスト読み込み

提出ファイルXLConnectで読み込み

```
library(XLConnect)
wb <- loadWorkbook(files[1])
```

ファイル名処理

- 番号と名前を抽出
- 全角の場合が多い
- １０１８０５１４５８石田.xlsx

文字処理

stringrパッケージ, stringiパッケージ

```
   <- stri_trans_nfkc_casefold(
"         .xlsx") %>%
    str_extract(
    "\\d{8,10}\\p{Han}*.xlsx$")
```

```
[1] "1018051458  .xlsx"
```

シートとセル

式を取得

```
A11 <- getCellFormula(wb,
           "Sheet1",
```

```
           #
           11, 1)
```

出力用データ

```
seiseki <- tibble(
  Id =   {  } ,
  Name = { } ,
  A11 =  {  } ,
  B11  = {   } ,
  C11 = {   } )
```

式の採点

if_else()：正答ならば1さもなければ0

```
seiseki <- seiseki %>%
  mutate(
    A11a = if_else(
      A11 == "AVERAGE(A2:A10)",
    1,  #
    0 ) #
```

評価の重み

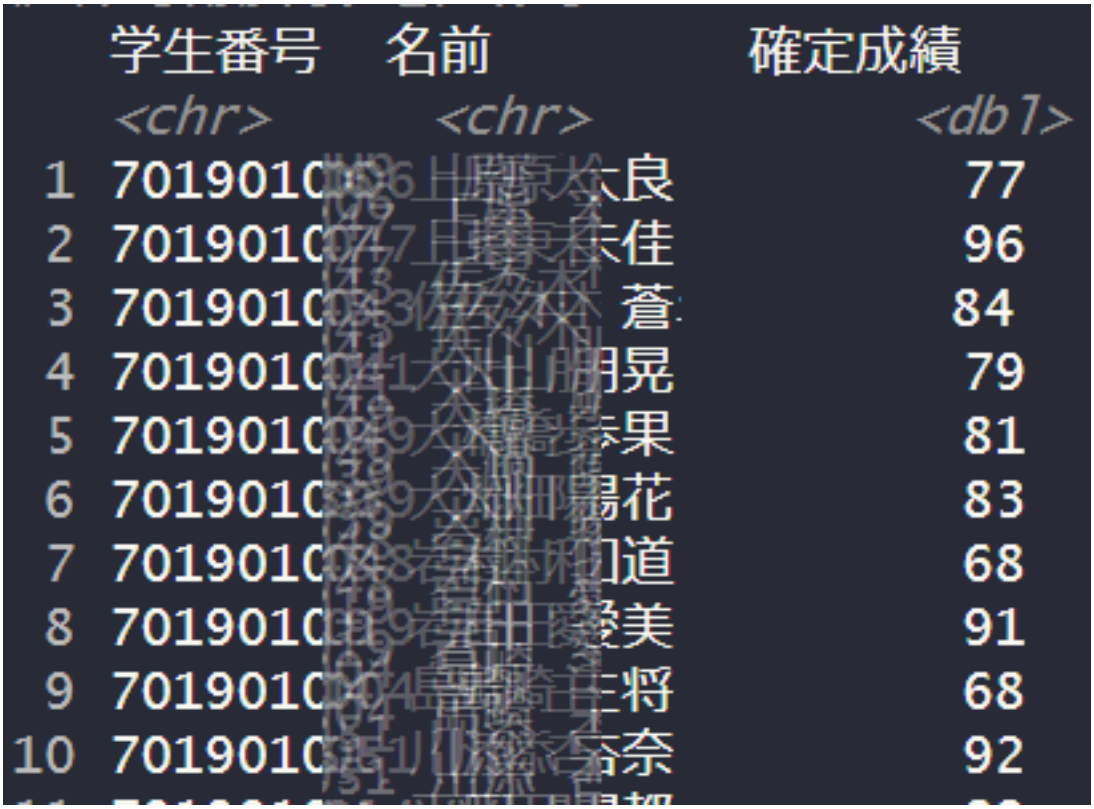
A11は38点、B11は36、C11は26点

```
seiseki <- seiseki %>%
  mutate(A11a = A11 * 38,
         B11a = B11 * 36,
         C11a = C11 * 26,)
```

評価の決定

```
seiseki %>% mutate(
     =
      rowSums(select_if(., is.numeric)))
# seiseki %>% rowwise %>%
#  mutate(  =sum(A11a, B11a, C11a)) %>%
#     select(  ,  ,   )
```

採点ファイル



	学生番号	名前	確定成績
	<chr>	<chr>	<dbl>
1	7019010	良	77
2	7019010	佳	96
3	7019010	蒼	84
4	7019010	晃	79
5	7019010	果	81
6	7019010	陽花	83
7	7019010	道	68
8	7019010	美	91
9	7019010	将	68
10	7019010	奈	92

ファイルを結合

`left_join(()`

a

x1	x2
A	1
B	2
C	3

+

b

x1	x3
A	T
B	F
D	T

=

Mutating Joins

x1	x2	x3
A	1	T
B	2	F
C	3	NA

dplyr::**left_join(a, b, by = "x1")**

Join matching rows from b to a.

x1	x3	x2
A	T	1
B	F	2
D	T	NA

dplyr::**right_join(a, b, by = "x1")**

Join matching rows from a to b.

x1	x2	x3
A	1	T
B	2	F

dplyr::**inner_join(a, b, by = "x1")**

Join data. Retain only rows in both sets.

x1	x2	x3
A	1	T
B	2	F
C	3	NA
D	NA	T

dplyr::**full_join(a, b, by = "x1")**

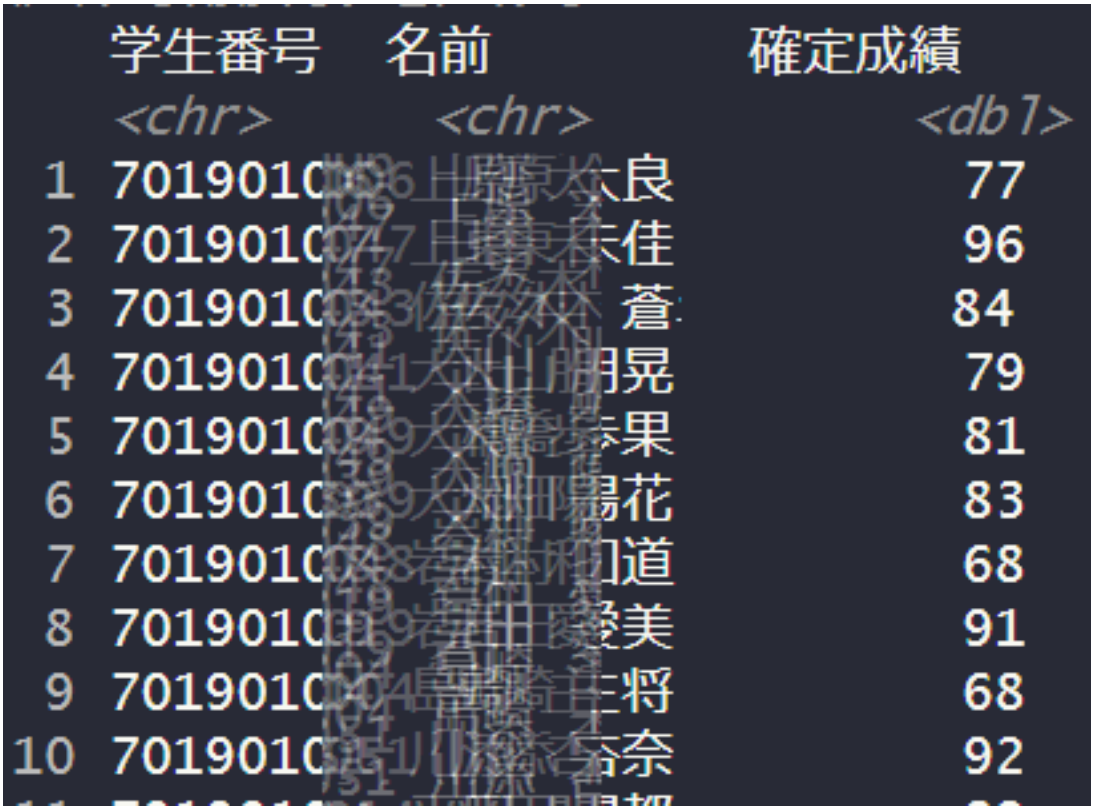
Join data. Retain all values, all rows.

原簿と結合

	A	B	C	D	E	F	G
1	No	年度	所属名	学生番号	氏名	性別	確定した評価
2	1	2018	地球科学	7019005101	実	女	
3	2	2018	地球科学	7019005303	未	女	
4	3	2018	地球科学	7019005401	さと	女	
5	4	2018	地球科学	7019005504	優	女	
6	5	2018	地球科学	7019005605	里	女	
7	6	2018	地球科学	7019005707		男	
8	7	2018	地球科学	7019005807	太	男	
9	8	2018	地球科学	7019005909		女	
10	9	2018	地球科学	7019006012	華	女	
11	10	2018	地球科学	7019006103	法	男	
12	11	2018	地球科学	7019006211	優	女	
13	12	2018	地球科学	7019006308	尋	女	
14	13	2018	地球科学	7019006407	里	女	
15	14	2018	地球科学	7019006506		男	
16	15	2018	地球科学	7019006608	菜	女	
17	16	2018	地球科学	7019006704	咲	女	
18	17	2018	地球科学	7019006802	衣	女	
19	18	2018	地球科学	7019006909		男	
20	19	2018	地球科学	7019007003	弥	男	

left_join

```
genbo <- genbo %>%
  left_join(seiseki,
    by = "  ")
```



	学生番号	名前	確定成績
	<chr>	<chr>	<dbl>
1	7019010	良	77
2	7019010	佳	96
3	7019010	蒼	84
4	7019010	晃	79
5	7019010	果	81
6	7019010	陽花	83
7	7019010	道	68
8	7019010	美	91
9	7019010	将	68
10	7019010	奈	92

join

単語頻度表からストップワード削除

```
stops <- tibble(
 TERM=c(" "," "))
stops
```

```
# A tibble: 2 x 1
  TERM
  <chr>
1
2
```

```
df %>%
 anti_join(stops)    %>% head()
```

```
  TERM   POS1                    POS2  1  2  3
1                                0     0     2
2                               0     1     0
3                               1     1     0
4                              *     0     1     0
5                   0     1     1
6                                0     0     3
```

グラフ作成

日本語

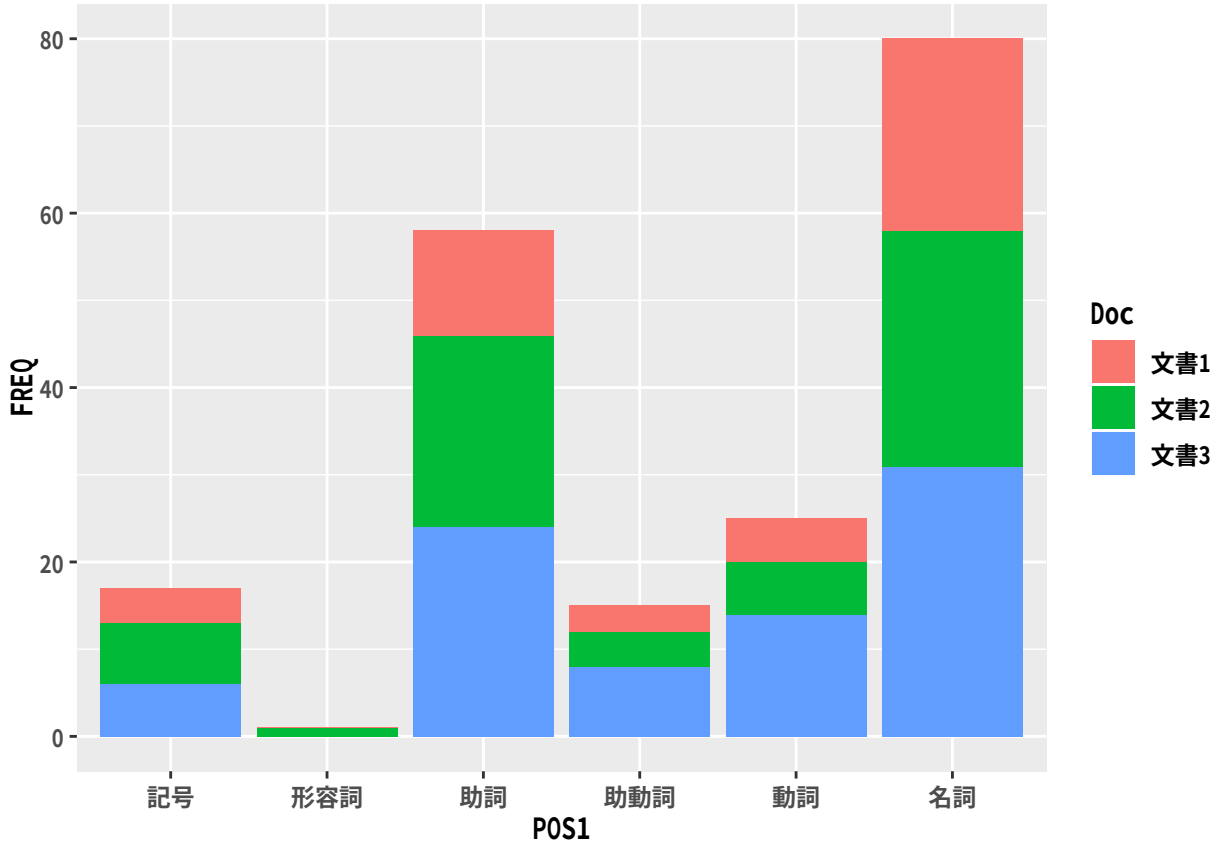
Macでは必須

```
source("http://rmecab.jp/R/Rprofile.R")
```

http://rmecab.jp/R/dot.Rprofile.txt

ggplot2

```
p <- doc %>%
  ggplot(aes(POS1,
             FREQ,
             fill=Doc)) +
  geom_bar(stat =
             "identity")
```



gapminderパッケージ

gapminder

tidyなデータ

横型はNG

```
# A tibble: 142 x 14
```

```
   country continent `1952` `1957` `1962` `1967` `1972` `1977` `1982`
   <fct>   <fct>      <dbl>  <dbl>  <dbl>  <dbl>  <dbl>  <dbl>  <dbl>
 1 Afghan~ Asia        28.8   30.3   32.0   34.0   36.1   38.4   39.9
 2 Albania Europe      55.2   59.3   64.8   66.2   67.7   68.9   70.4
 3 Algeria Africa      43.1   45.7   48.3   51.4   54.5   58.0   61.4
 4 Angola  Africa      30.0   32.0   34     36.0   37.9   39.5   39.9
 5 Argent~ Americas    62.5   64.4   65.1   65.6   67.1   68.5   69.9
 6 Austra~ Oceania     69.1   70.3   70.9   71.1   71.9   73.5   74.7
 7 Austria Europe      66.8   67.5   69.5   70.1   70.6   72.2   73.2
 8 Bahrain Asia        50.9   53.8   56.9   59.9   63.3   65.6   69.1
 9 Bangla~ Asia        37.5   39.3   41.2   43.5   45.3   46.9   50.0
10 Belgium Europe      68     69.2   70.2   70.9   71.4   72.8   73.9
# ... with 132 more rows, and 5 more variables: `1987` <dbl>,
#   `1992` <dbl>, `1997` <dbl>, `2002` <dbl>, `2007` <dbl>
```

基本的作画の流れ

- ggplot で初期化
- aes で変数を指定
- geom_でグラフの種類

Gapminder

```
gap <- gapminder %>%
  filter(country %in%
    c("Japan", "China", "Korea, Rep."))
```

（簡単のため国を3つに絞る）

ggplo初期化

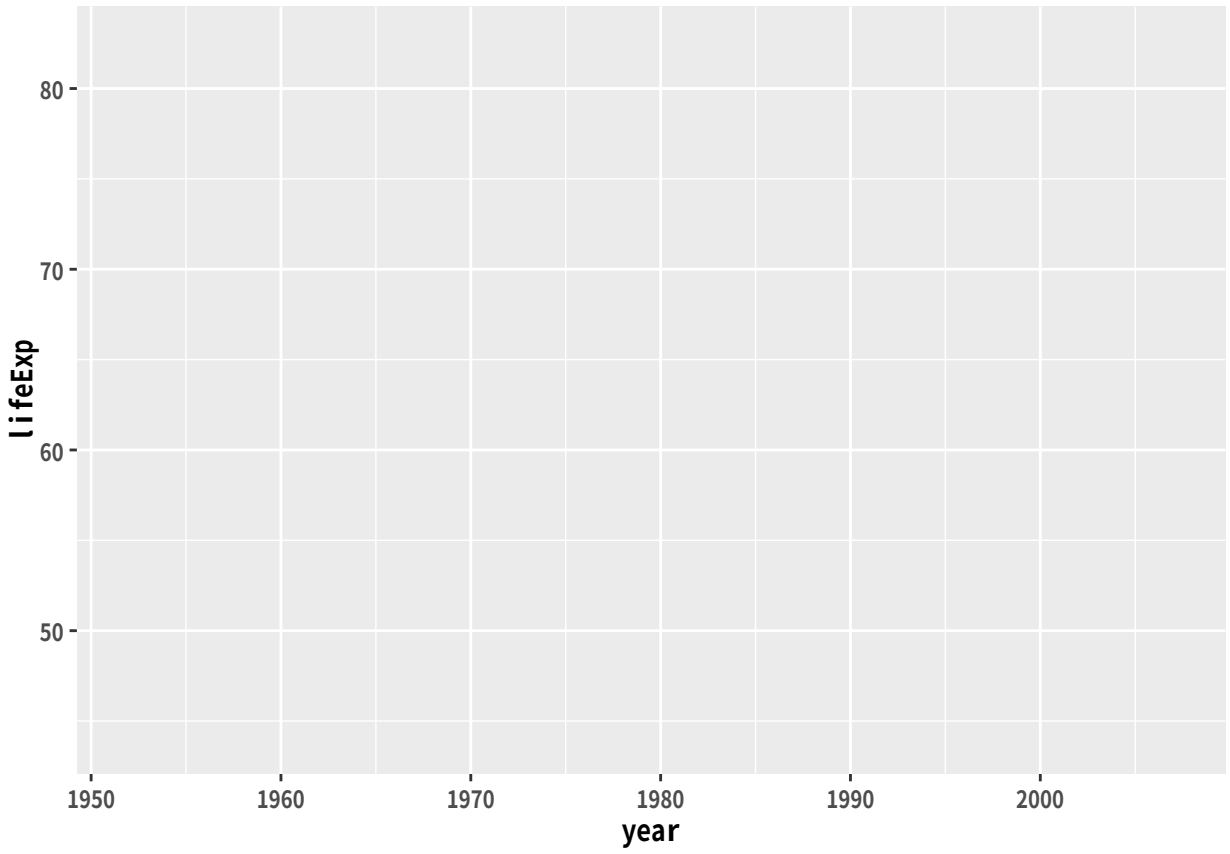
データセットをggplot2に指定

```
library(ggplot2)
p <- gap %>%
  ggplot()
#p<-ggplot(gap)
```

Aesthetic mappings

データと軸やカラーを対応させる

```
p <- p +
  aes(x =year,
      y =lifeExp,
      size =pop,
      col =country)
```



土台とデータ対応

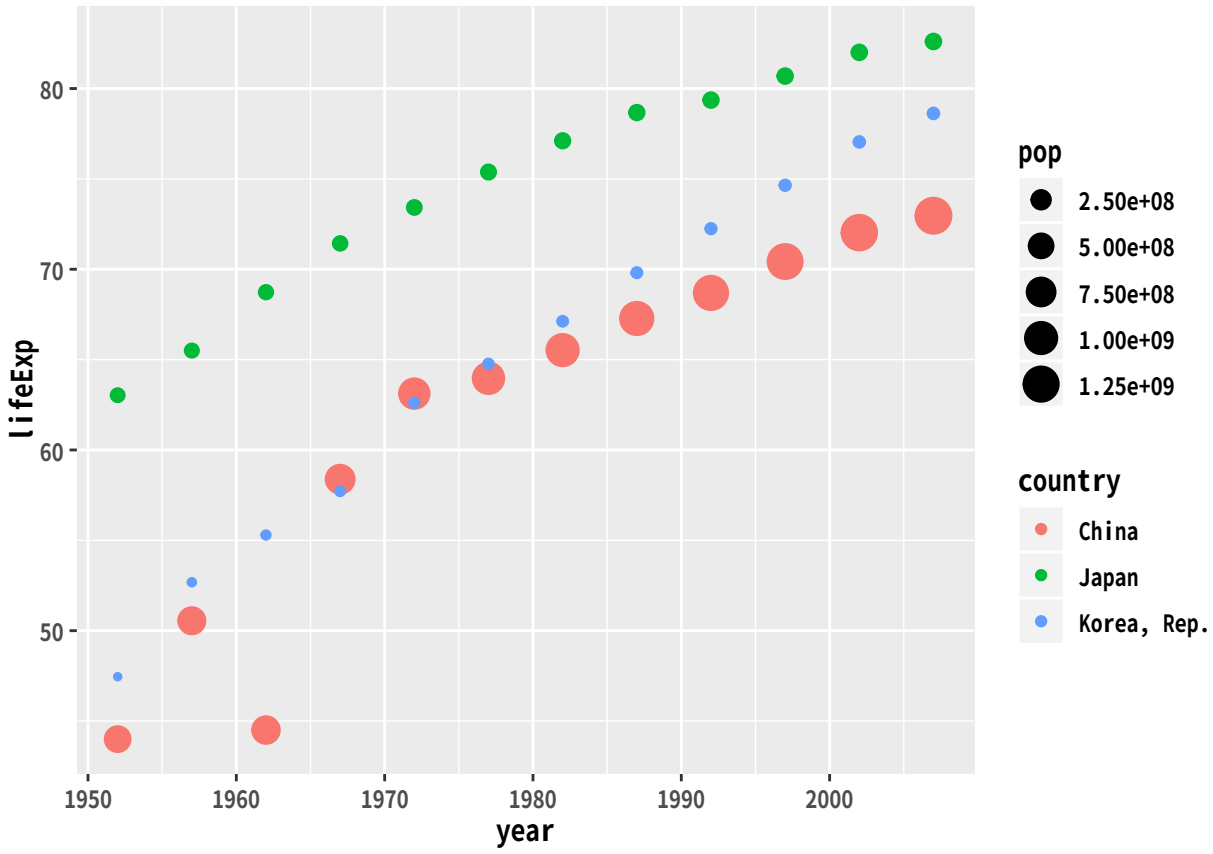
ggplot() + aes(…) == ggplot(aes (…))

```
gap %>%
  ggplot(aes(x = year, y = lifeExp,
         size = pop,col = country))
```

aes で水準ごとの指定

散布図

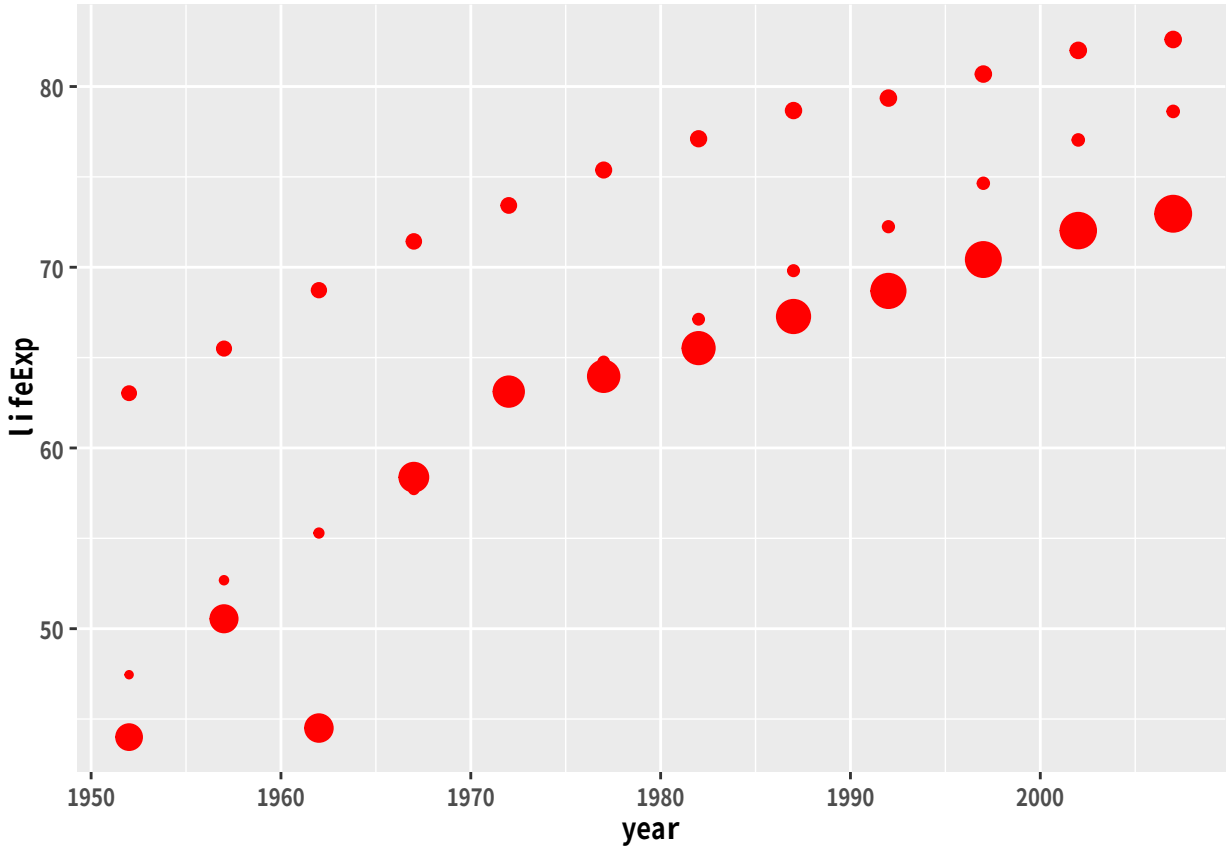
```
p <- p+geom_point()
```



aesの外で

色を指定

```
p0 <- gap %>%
  ggplot(aes(
   x = year,
   y = lifeExp,
   size = pop))
p0<-p0+geom_point(
    show.legend =
    FALSE,
    col ="red") #
```



geom_族

geom_bar

geom_point

geom_line

geom_boxplot

バーチャート

各年の個人GDP平均

```
gap %>%
  group_by(year) %>%
  summarise(AVG = mean(gdpPercap))
```

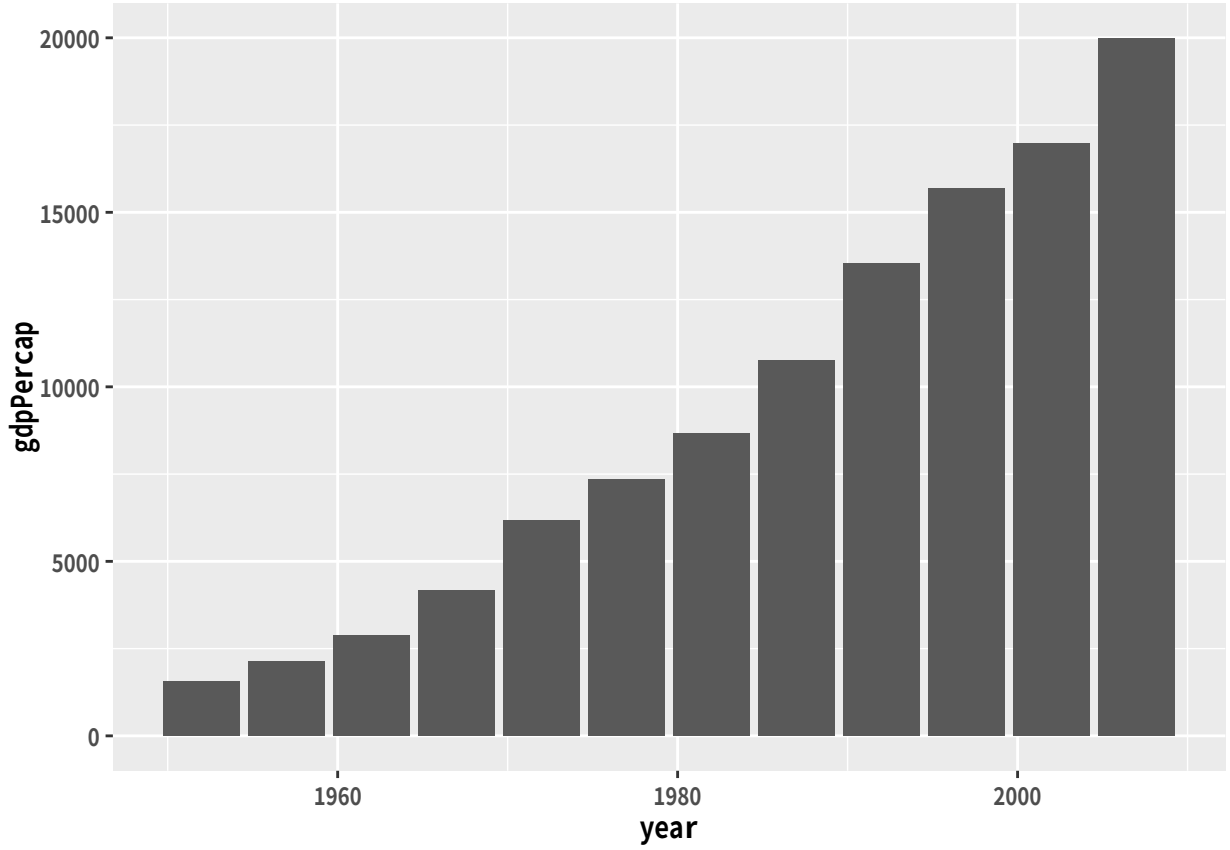
```
# A tibble: 12 x 2
    year    AVG
   <int>  <dbl>
 1  1952  1549.
 2  1957  2127.
 3  1962  2867.
 4  1967  4163.
 5  1972  6162.
 6  1977  7336.
 7  1982  8656.
```

```
 8  1987 10763.
 9  1992 13528.
10  1997 15700.
11  2002 16986.
12  2007 19988.
```

summary

stat = “summary”

```
p <- gap %>%
  ggplot(aes(year,
    gdpPercap))
p <- p +
  geom_bar(stat =
    "summary",
    fun.y = "mean")
```



geom_bar

もし集計済みデータだった場合

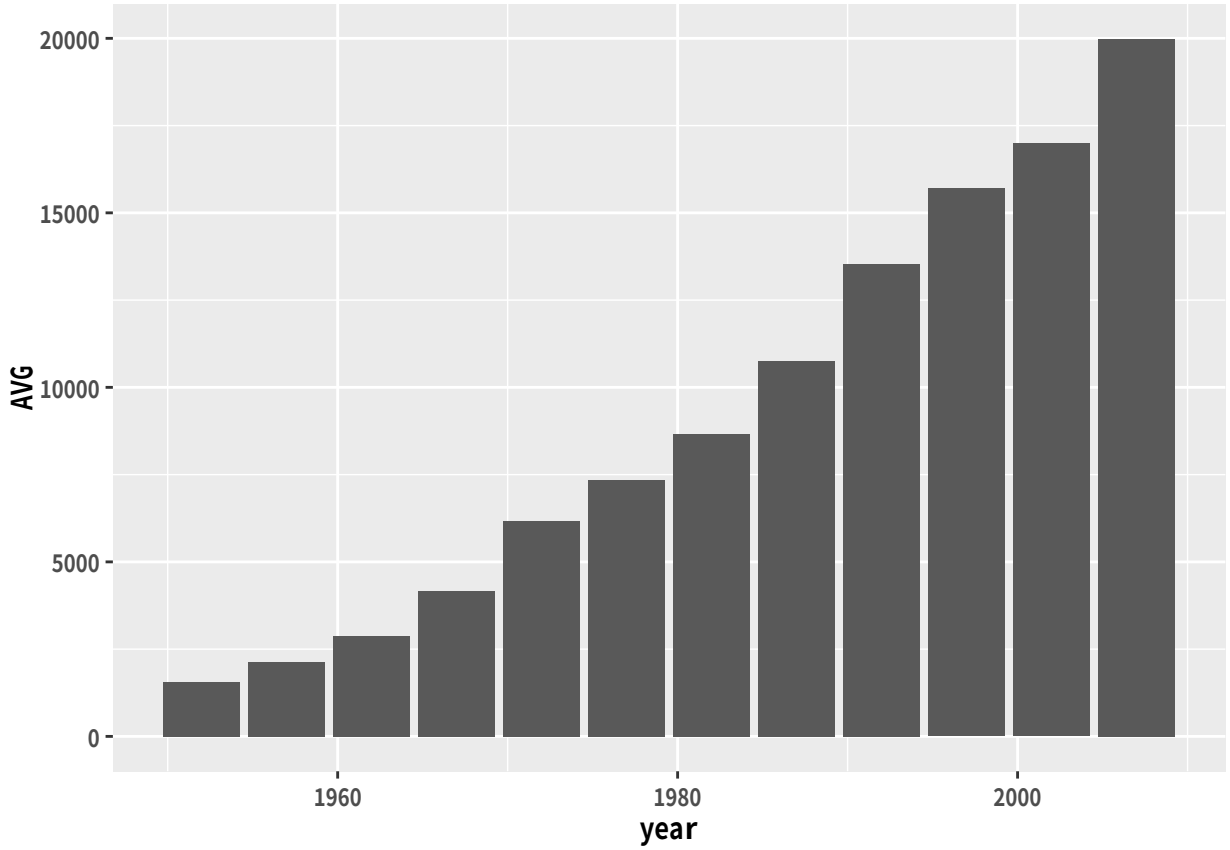
```
# A tibble: 12 x 2
    year    AVG
   <int>  <dbl>
 1  1952  1549.
 2  1957  2127.
```

```
 3  1962  2867.
 4  1967  4163.
 5  1972  6162.
 6  1977  7336.
 7  1982  8656.
 8  1987 10763.
 9  1992 13528.
10  1997 15700.
11  2002 16986.
12  2007 19988.
```

geom_bar

stat ="identity"

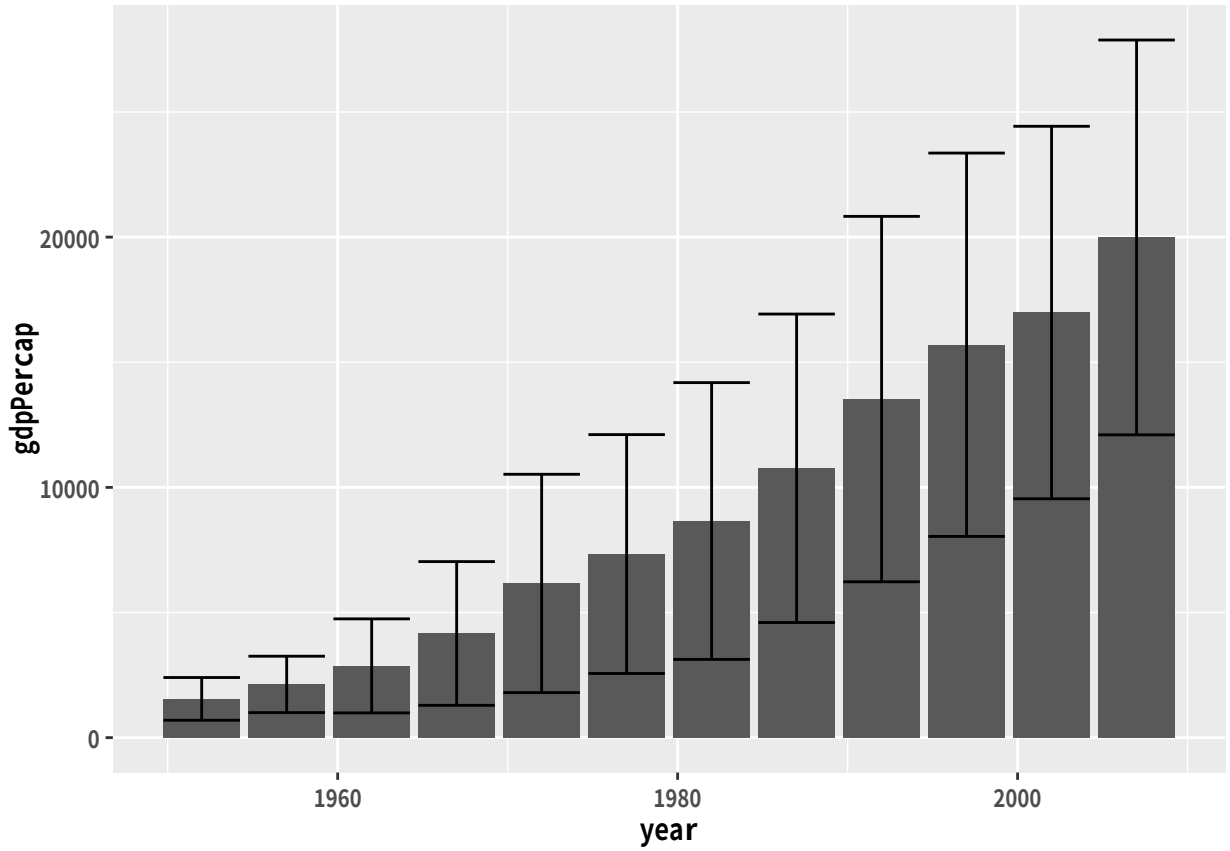
```
p <- gap_avg %>%
  ggplot(
   aes(year, AVG))
p <- p +
  geom_bar(
  stat ="identity")
```



stat_summary

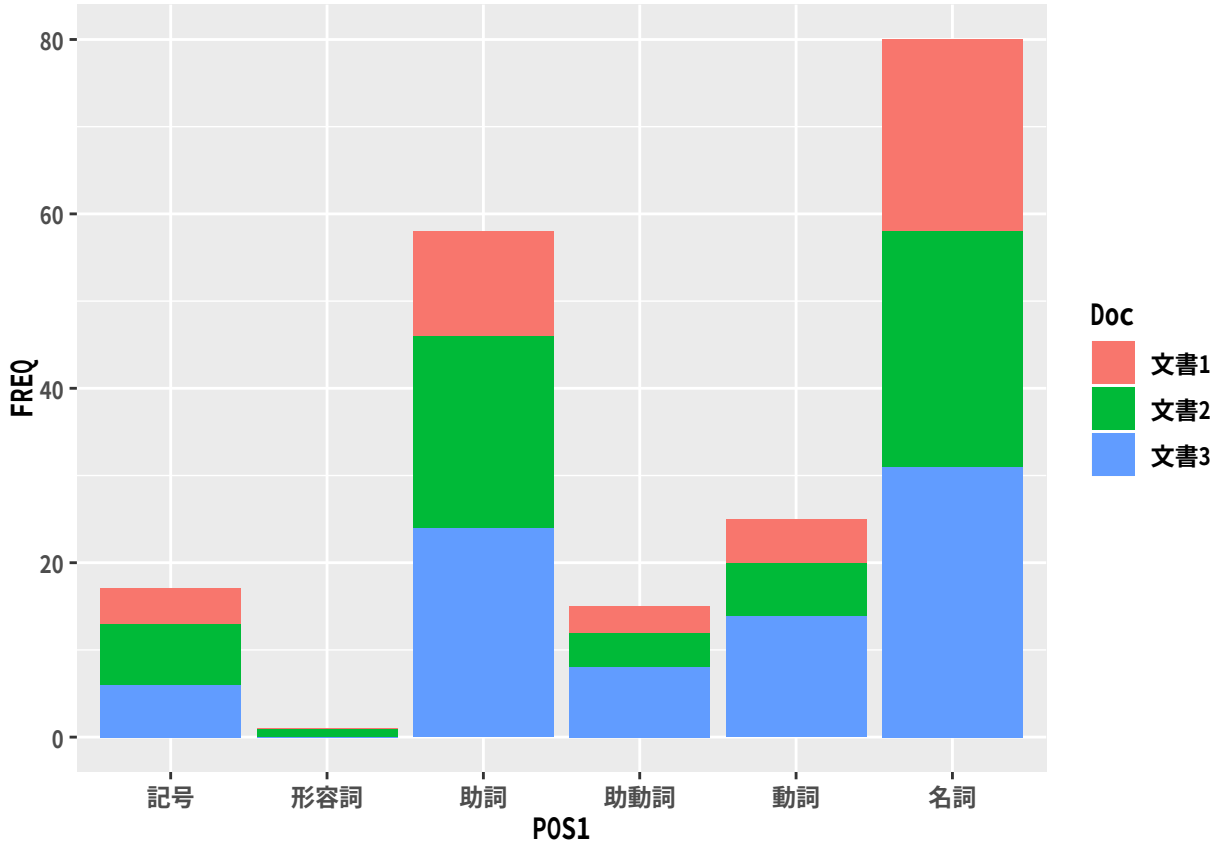
エラーバー

```
p <- p +
stat_summary(
  geom = "bar",
  fun.y = "mean") +
stat_summary(
  geom = "errorbar",
  fun.data =
    "mean_se")
```



品詞の頻度

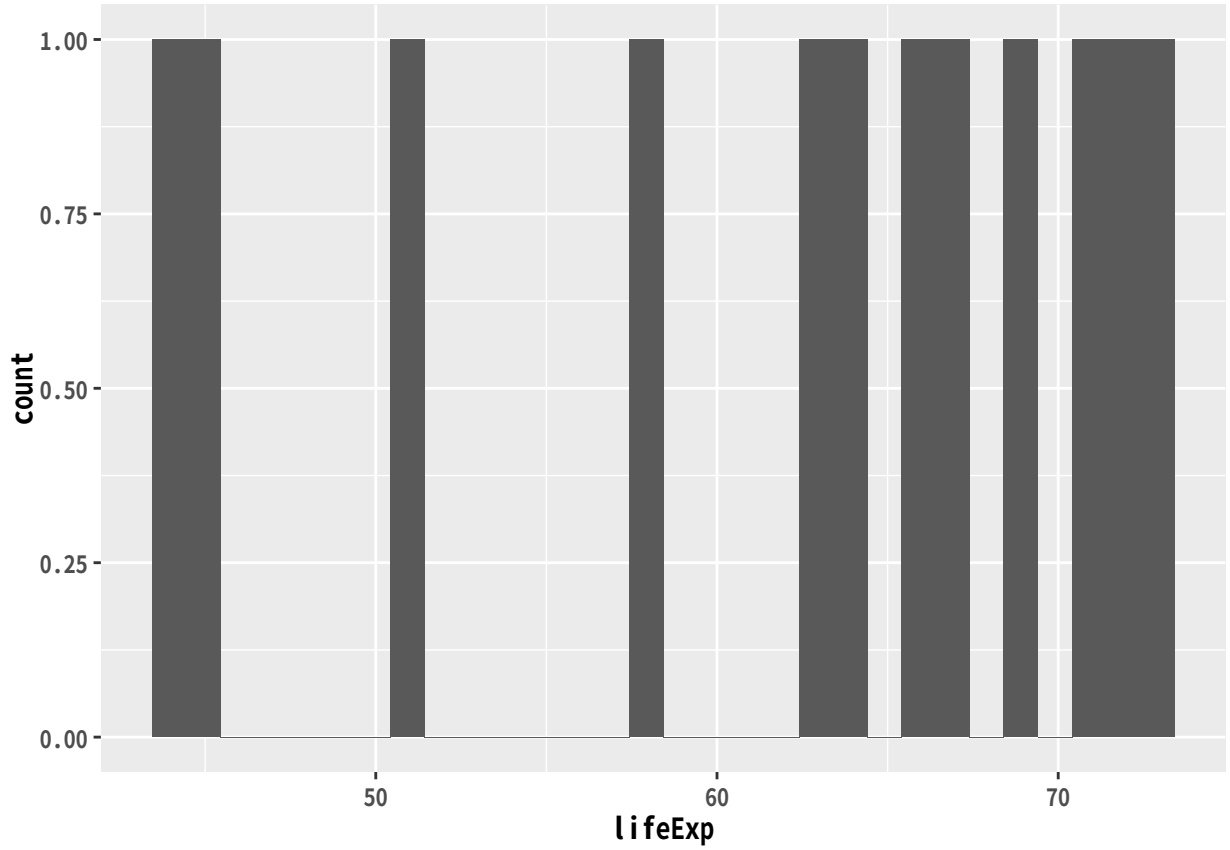
```
p <- doc %>%
  ggplot(aes(POS1,
             FREQ,
         fill = Doc))
p <- p +
 geom_bar(
  stat="identity")
```



geom_histogram

中国人口

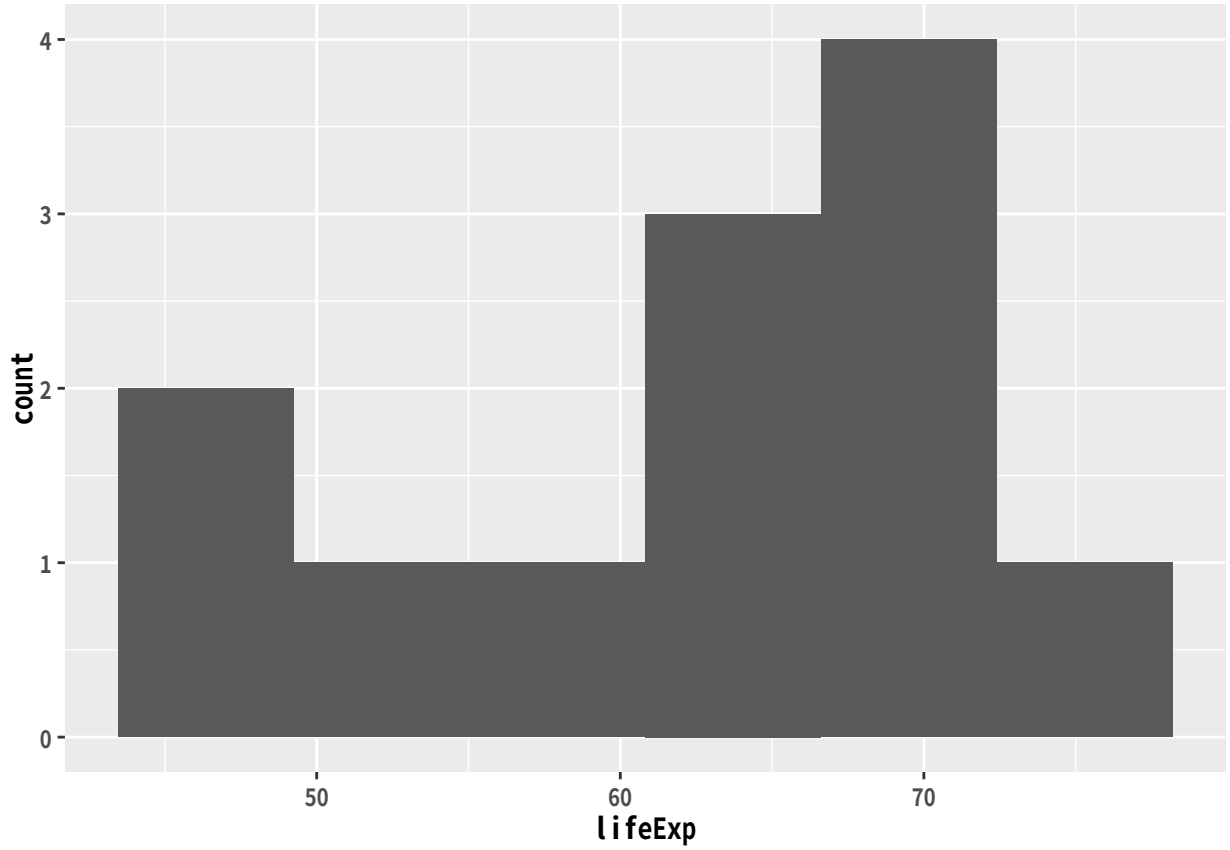
```
china <- gap %>%
 filter(
  country == "China")
p <- china %>%
  ggplot(
    aes(lifeExp))+
    geom_histogram()
```



binの指定

bins, binwidth

```
p <- china %>%
  ggplot(
    aes(lifeExp)) +
  geom_histogram(
    bins = 6)
```

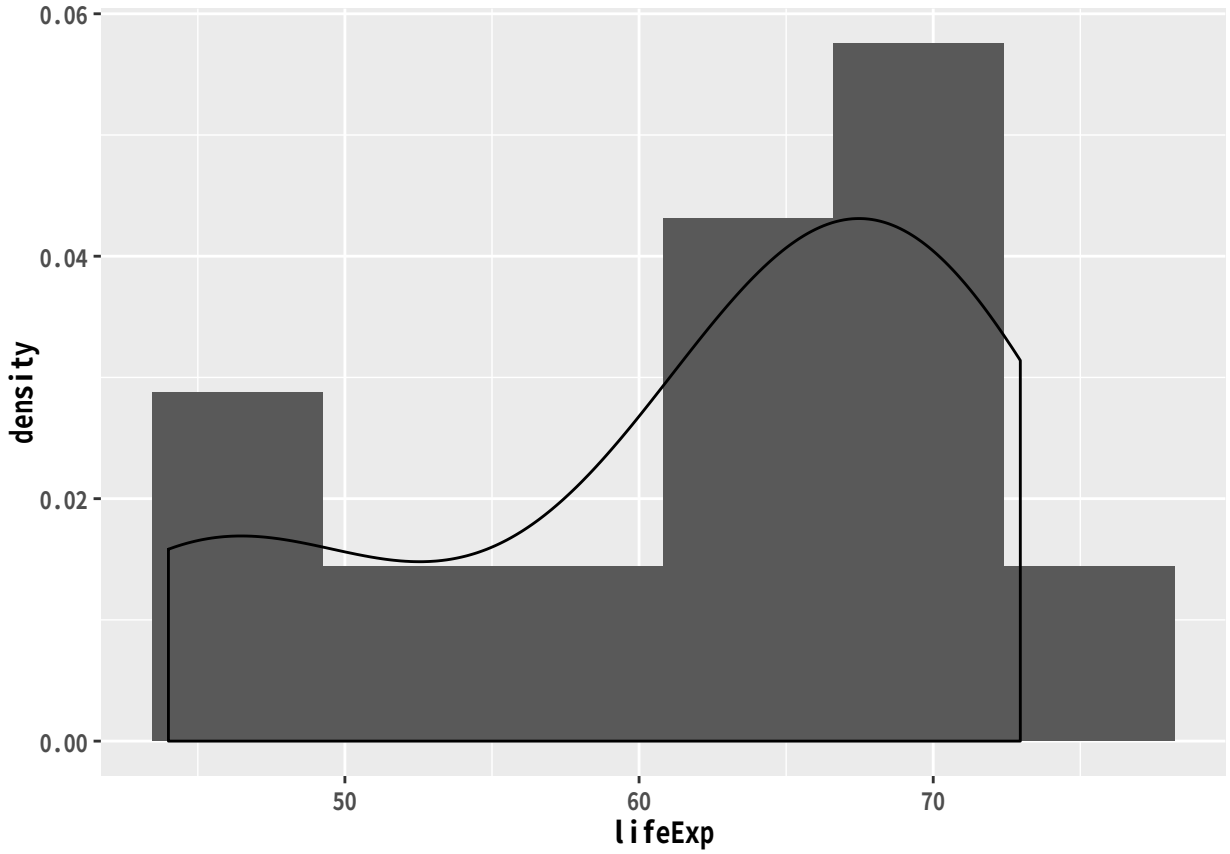


密度指定

..density..

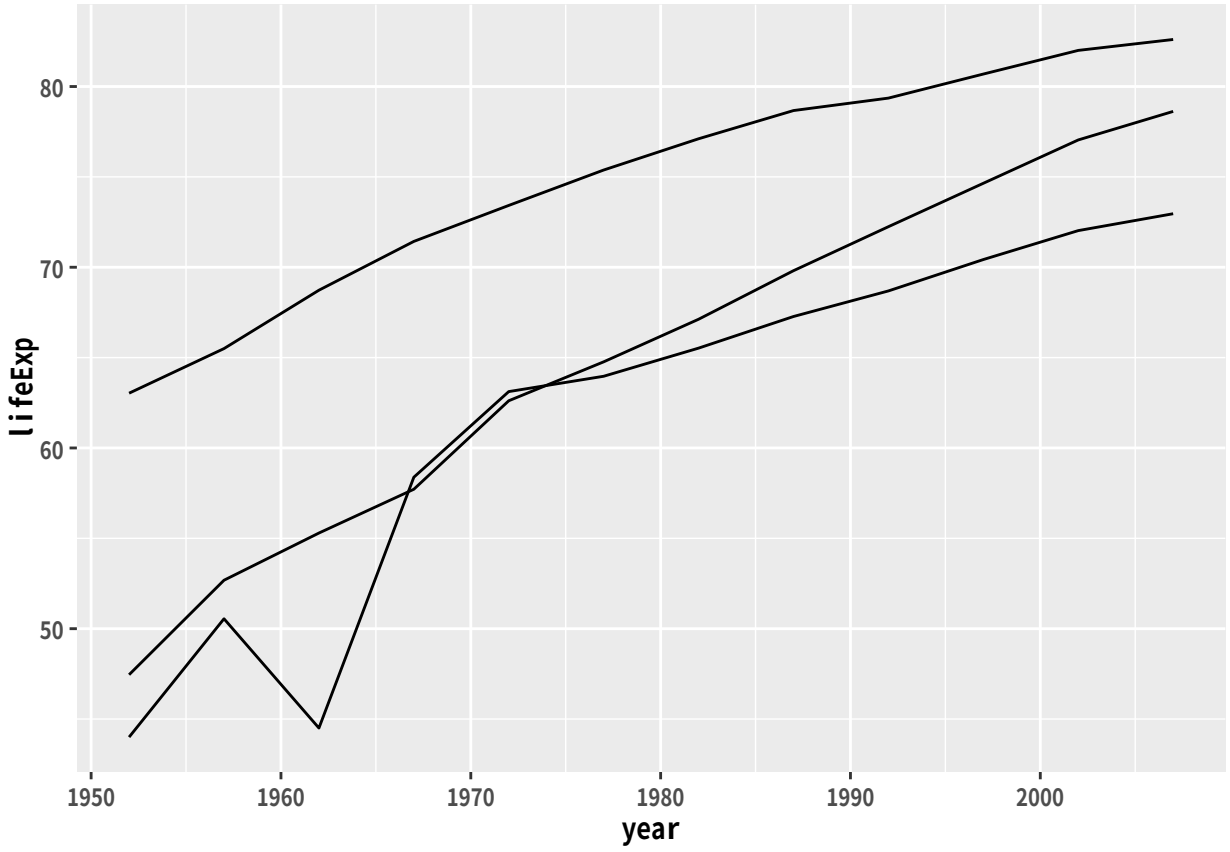
```
p <- china %>%
  ggplot(
    aes(lifeExp,
    y=..density..))+
  geom_histogram(
    bins = 6)
```

```
p + geom_density()
```



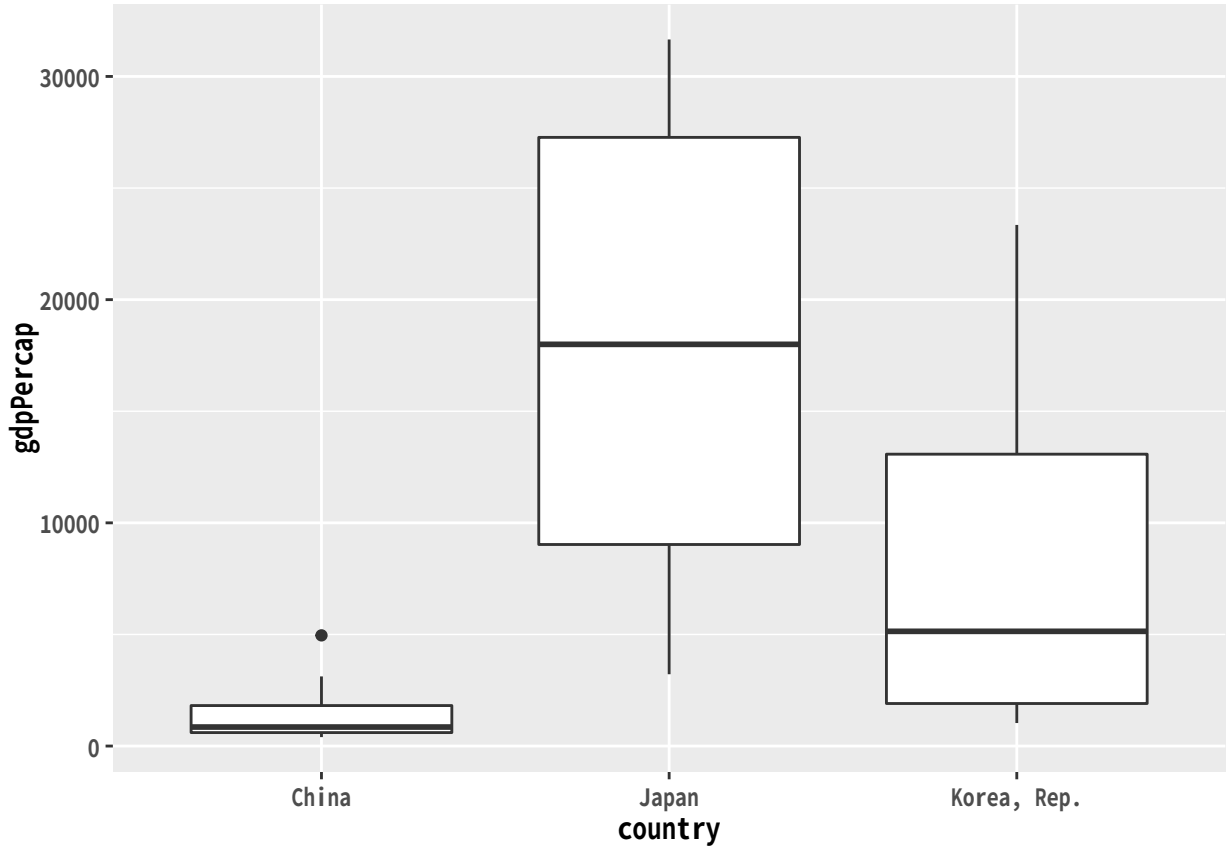
geom_line

```
p <- gap %>%
  ggplot(aes(year,
    lifeExp,
    group=country))
p <- p + geom_line()
```



geom_boxplot

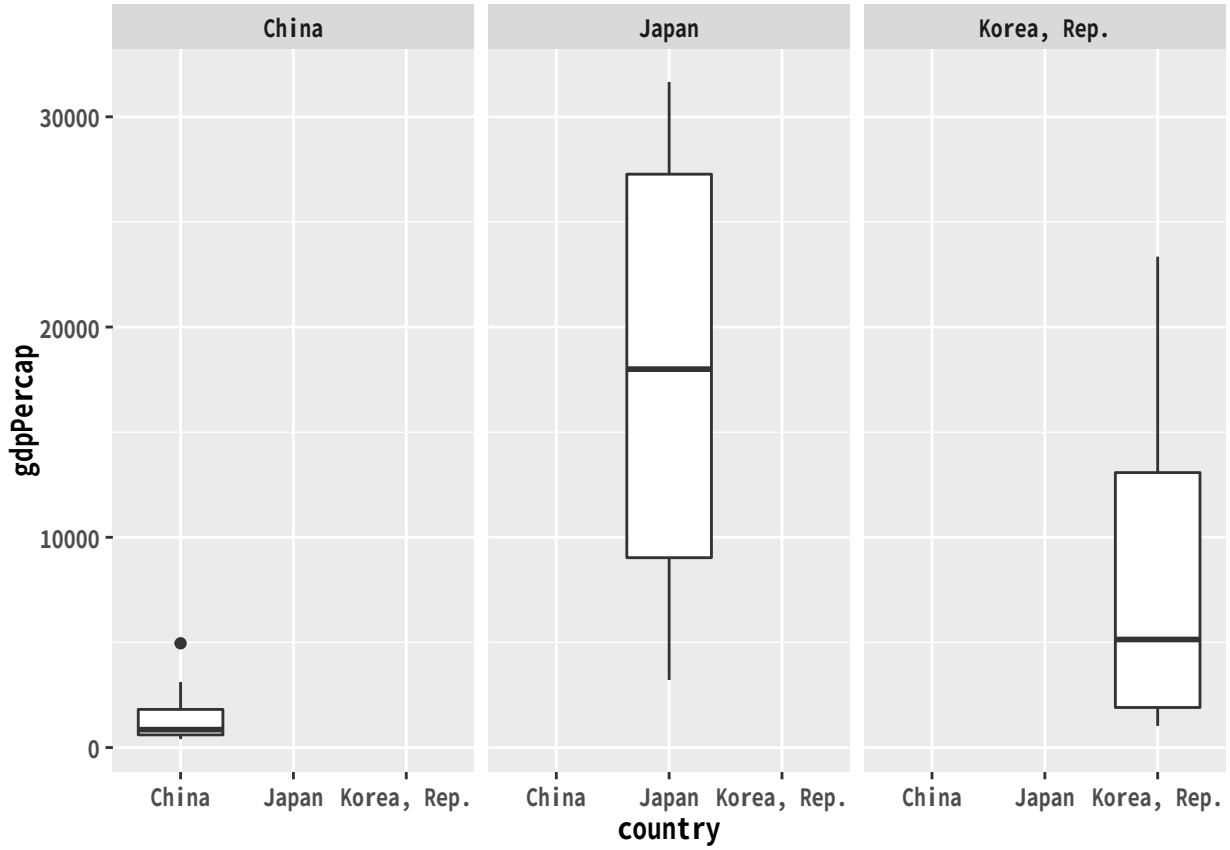
```
p <- gap %>%
  ggplot(
   aes(x=country,
       y=gdpPercap))
p <- p +
     geom_boxplot()
```



facet

`facet_grid`, `facet_wrap`

```
p_f <- p +
  facet_grid(
    . ~ country)
```

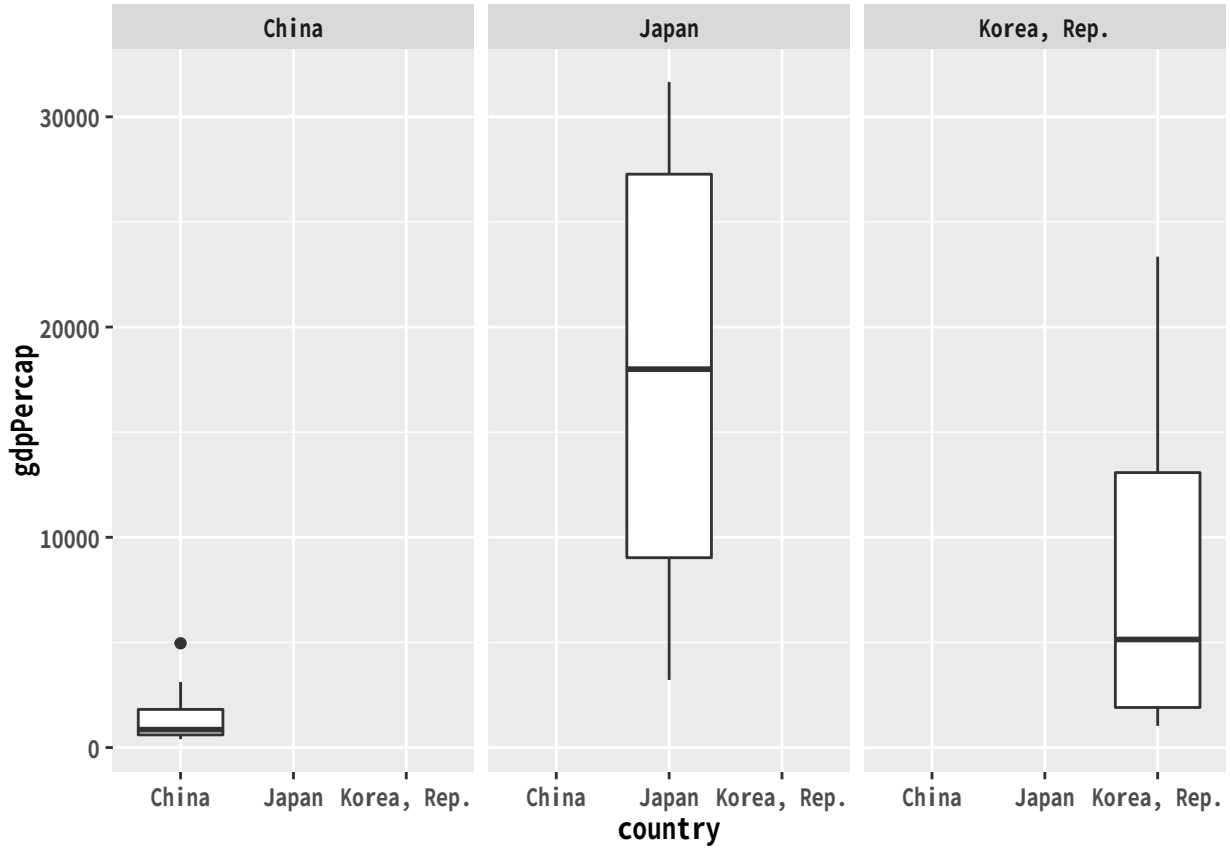


face_wrap

facet_grid(行[縦軸] ~ 列[横軸])

facet_wrap(~ 変数)

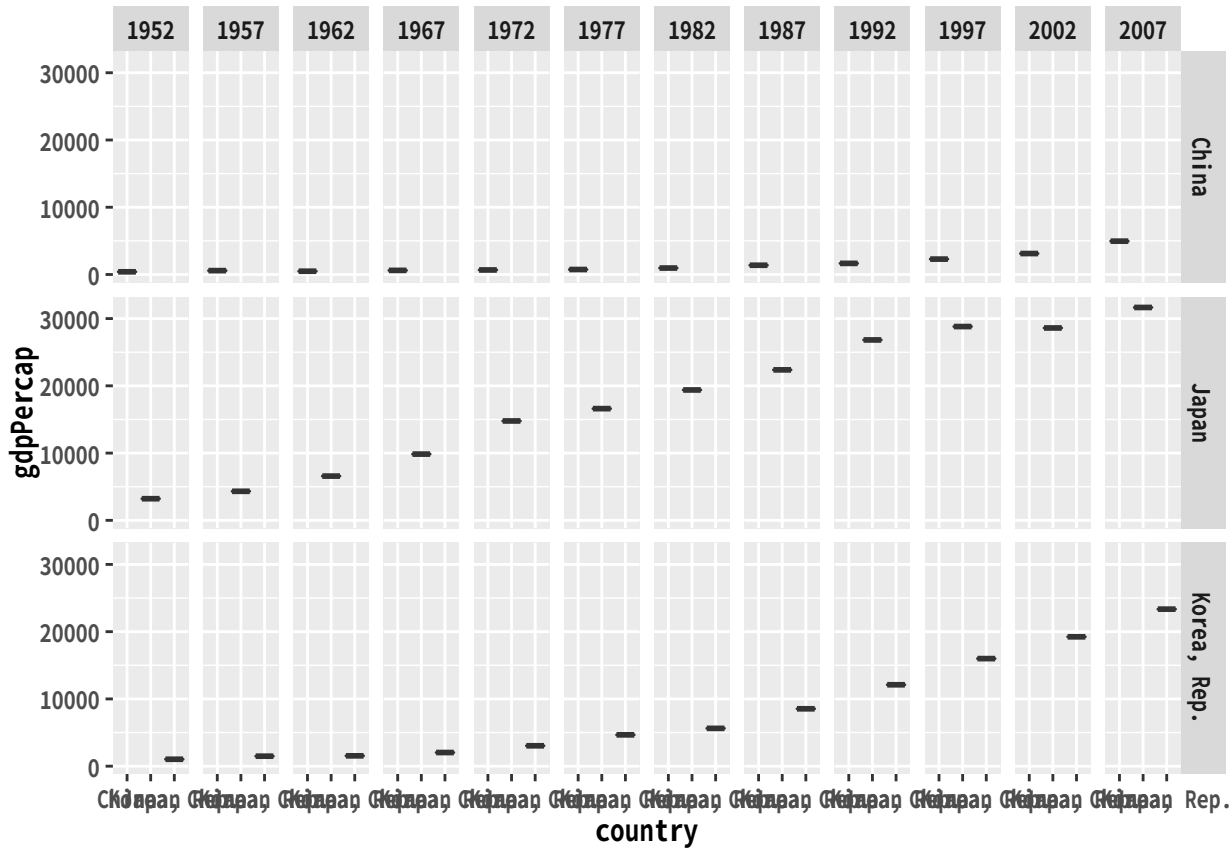
```
p +
  facet_wrap(
    ~ country)
```



grid or wrap

gridと wrapの違い

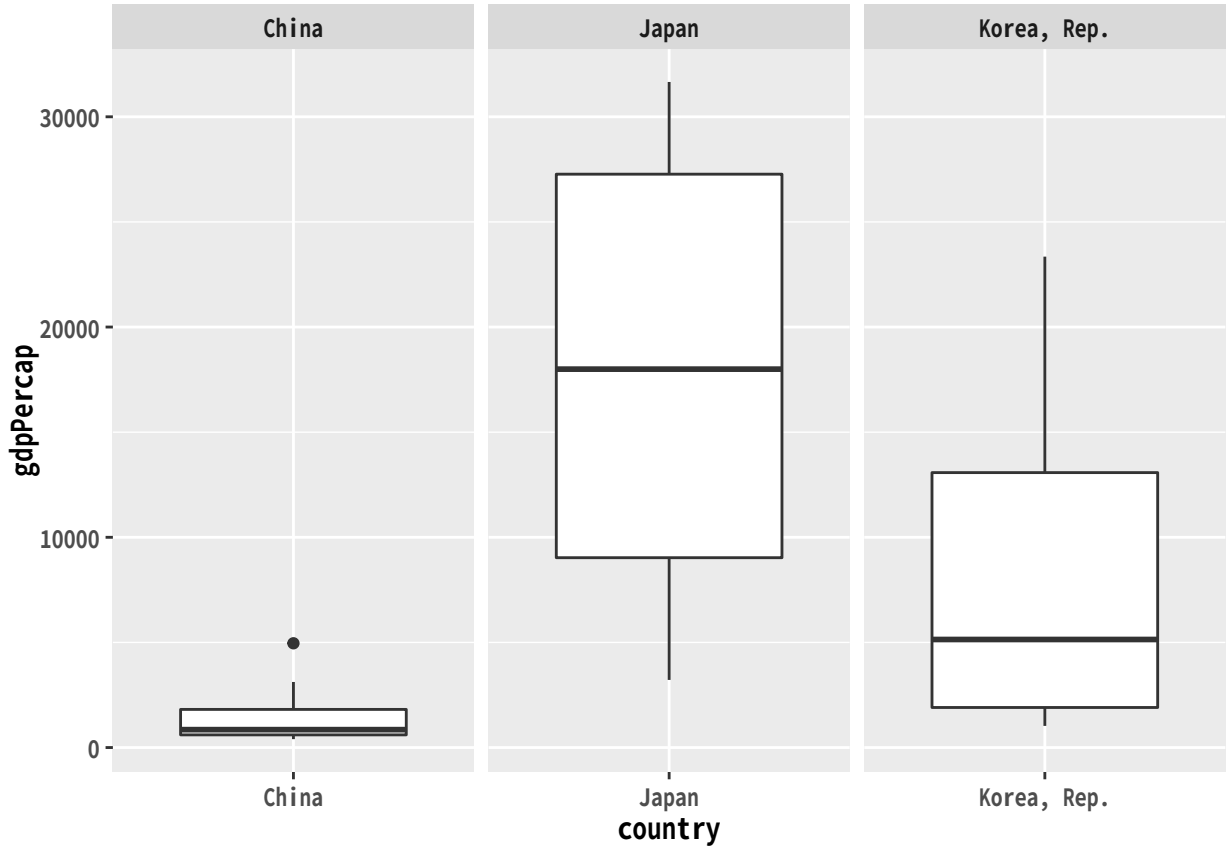
```
p + facet_grid(
  country~year)
```




```
p + facet_wrap(
  country~year)
```

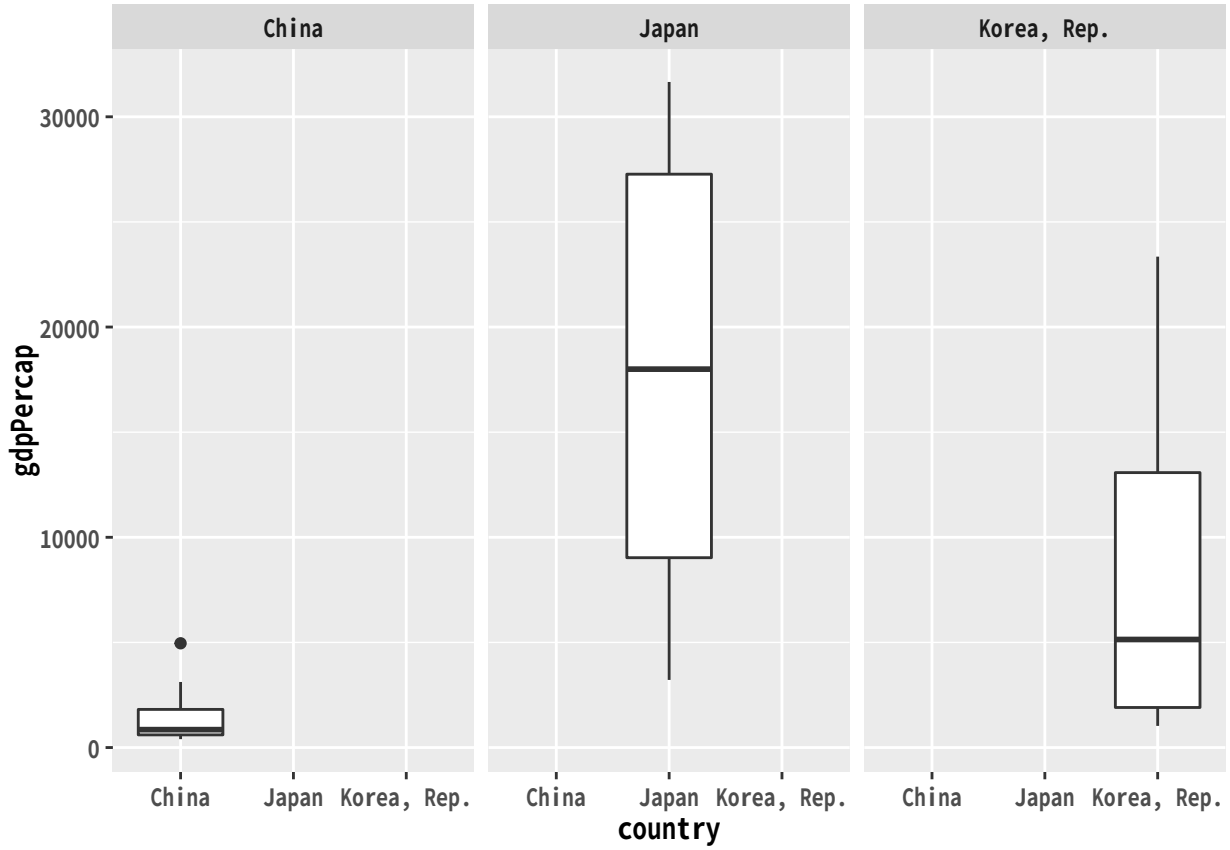
scales

```
p_f <- p +
  facet_grid(.
    ~ country,
    scales = "free")
```



no_scales

```
p_f <- p +
  facet_grid(.
    ~ country)
# "free"
```

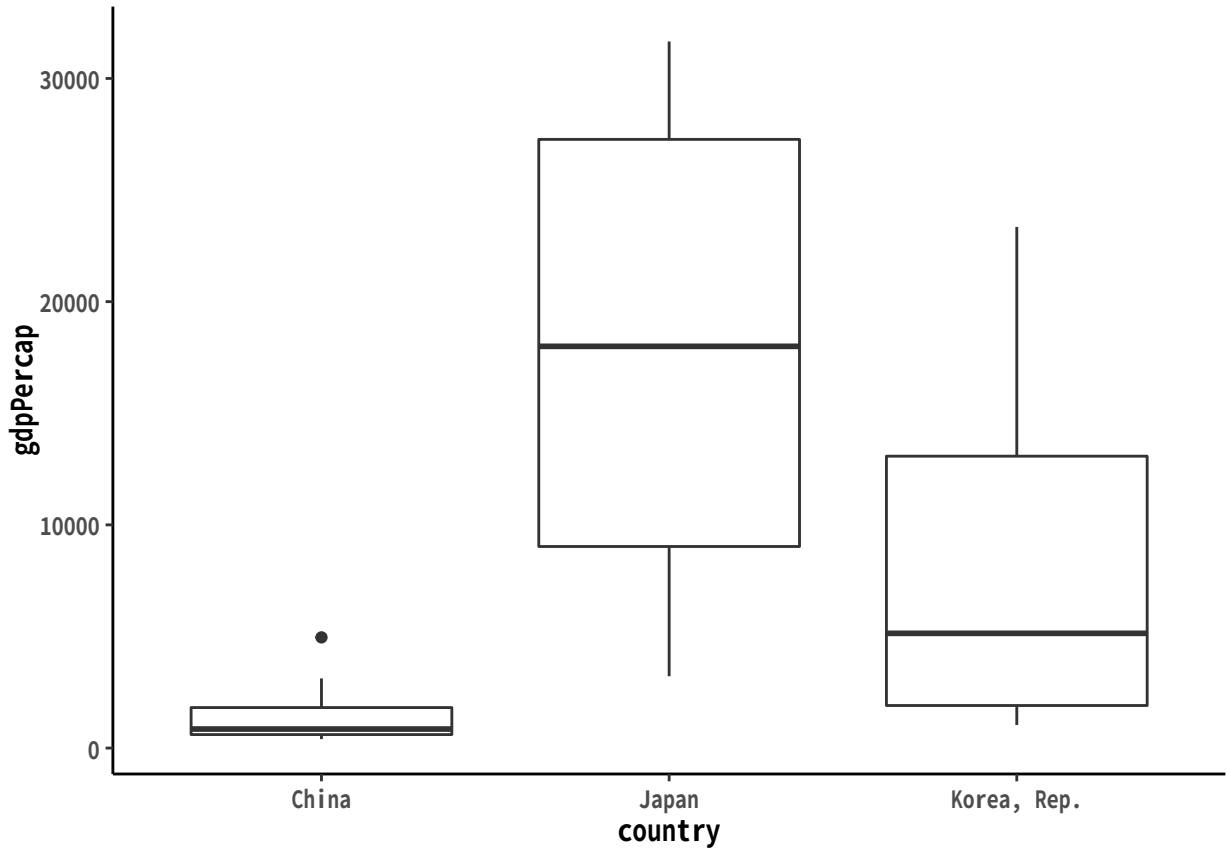


テーマ

背景を消したい theme_*

```
p <- p +
  theme_classic()
```

theme_bw() theme_light() theme_minimal()



referance

https://r4ds.had.co.nz/graphics-for-communication.html