

# Modern Data Science Methods for Educational Research

Creating Effective Data Visualization with R

Data Representation I:  
Visualizing “Amounts” & “Distributions”

ผศ.ดร.สิริวัฒน์ ครีสุทธิยการ

ภาควิชาวิจัยและจิตวิทยาการศึกษา  
คณะครุศาสตร์ จุฬาลงกรณ์มหาวิทยาลัย

February 18, 2023





## ชุดข้อมูลตัวอย่าง

ชุดข้อมูลที่ใช้เป็นตัวอย่างสำหรับเนื้อหาส่วนนี้คือ gapminder

```
1 #install.packages("gapminder")
2 library(gapminder)
3 head(gapminder, 3)
```

```
# A tibble: 3 x 6
  country   continent   year lifeExp      pop gdpPerCap
  <fct>     <fct>     <int>   <dbl>    <int>     <dbl>
1 Afghanistan Asia       1952     28.8  8425333    779.
2 Afghanistan Asia       1957     30.3  9240934    821.
3 Afghanistan Asia       1962     32.0  10267083   853.
```

## Visualizing Amounts

# Visualizing Amounts



ที่มา : Wike (2019)

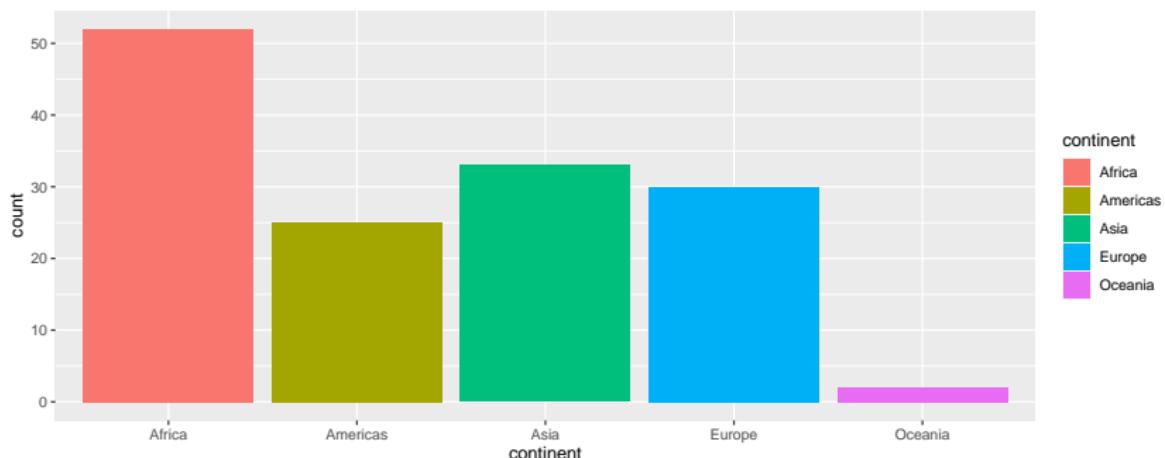
## Creating barplot using `geom_bar()`

ใช้สร้าง barplot ได้ทั้ง simple barplot, grouped barplot และ stacked barplot มีการกิวเมนท์สำคัญได้แก่

- ▶ `data` และ `mapping`
- ▶ `stat = "count"` หรือ `"identity"`
- ▶ `position = "stack"` หรือ `identity, dodge`
- ▶ `width`
- ▶ `orientation`
- ▶ `na.rm`

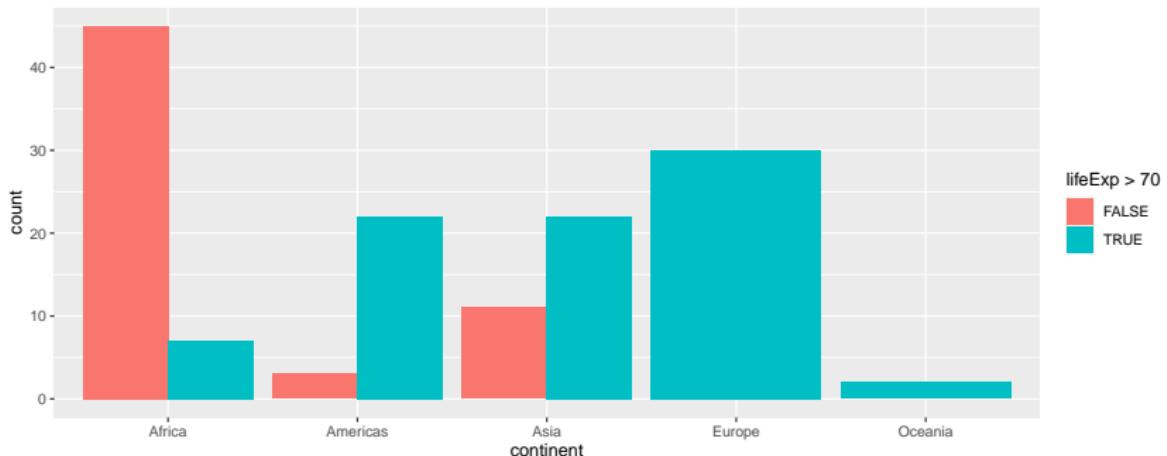
## Simple barplot

```
1 gapminder %>% filter(year == "2007") %>%
2   ggplot() +
3   geom_bar(aes(x = continent, fill=continent),
4             position = "identity")
```



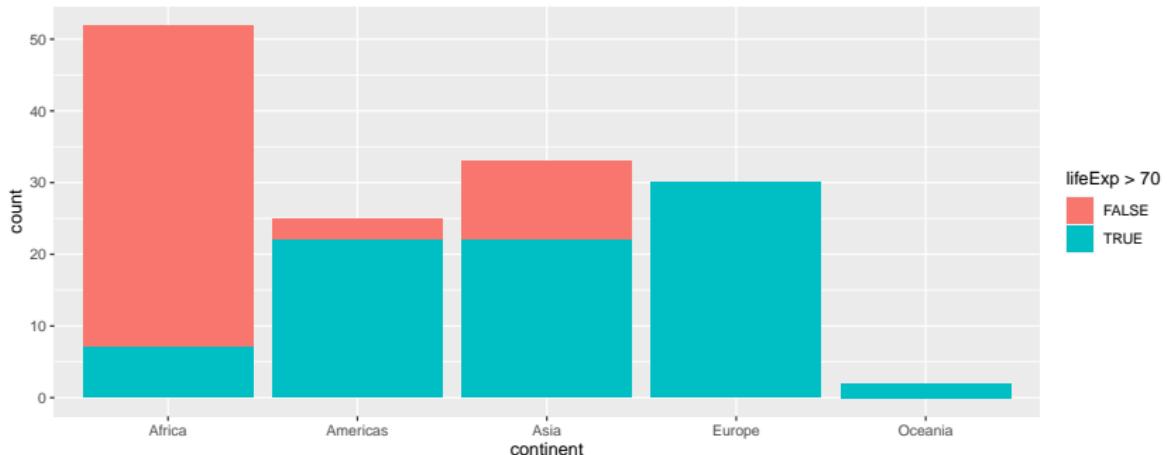
## Grouped barplot

```
1 gapminder %>% filter(year == "2007") %>%
2   ggplot()+
3   geom_bar(aes(x = continent, fill = lifeExp > 70),
4             position = "dodge")
```

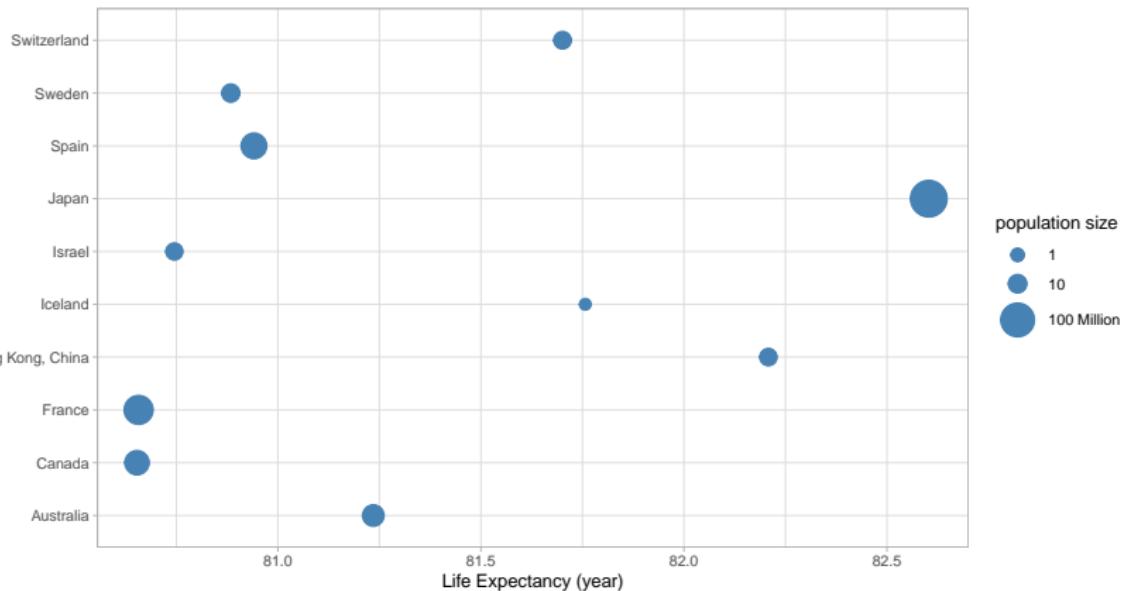


## Stacked barplot

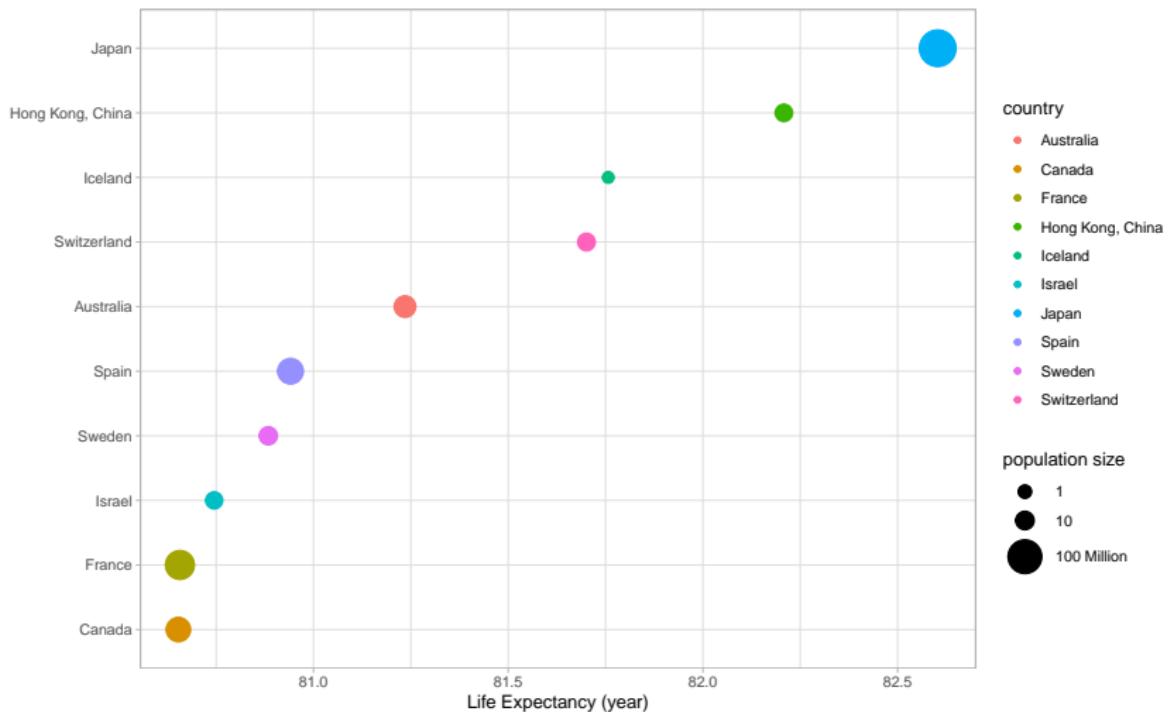
```
1 gapminder %>% filter(year == "2007") %>%
2   ggplot()+
3   geom_bar(aes(x = continent, fill = lifeExp > 70),
4             position = "stack")
```



## Creating Dotplot using `geom_point()`

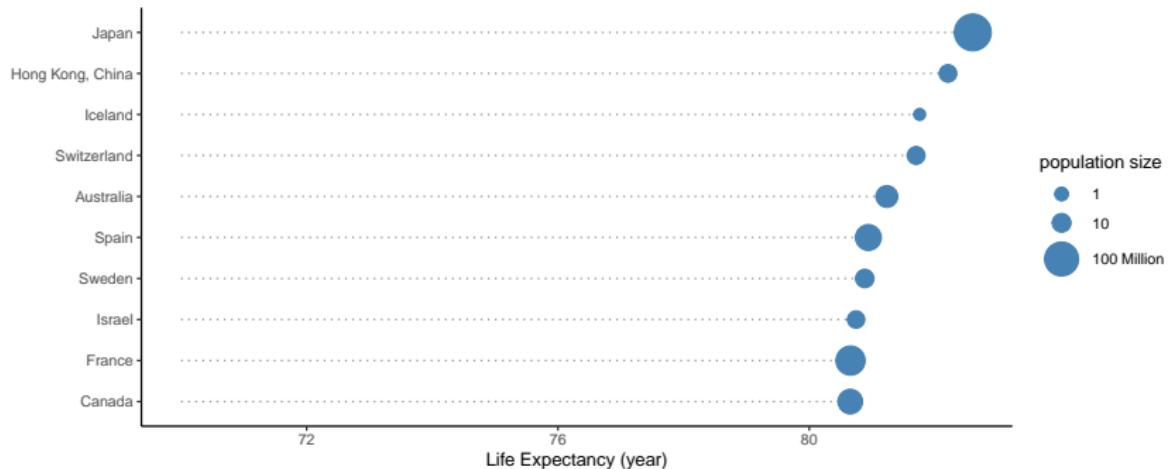


## Creating Dotplot using `geom_point()`



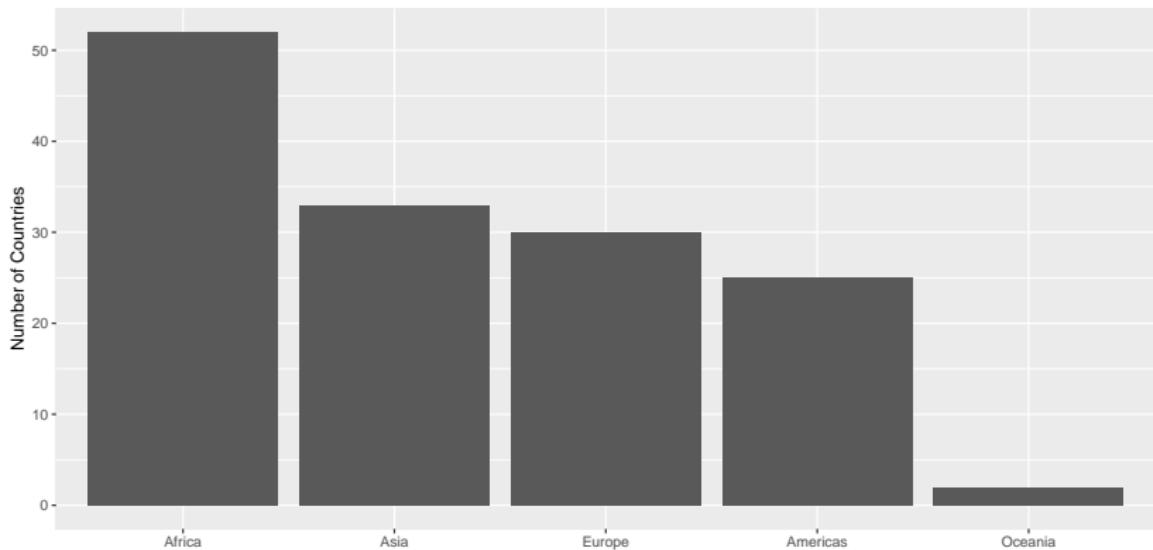
## Creating Lollipop plot

```
geom_segment(aes(x, y, xend, yend))
```

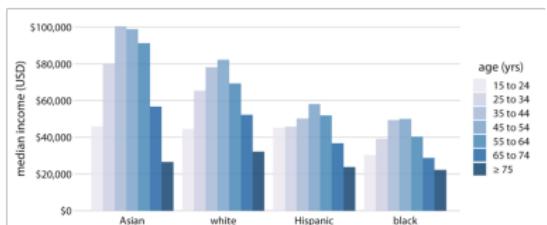
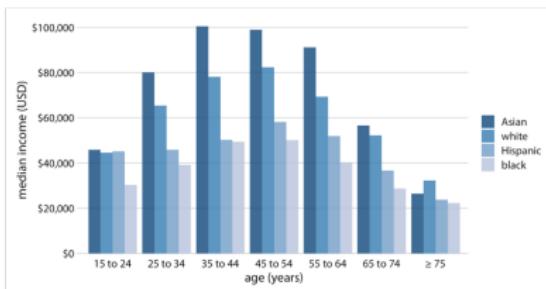
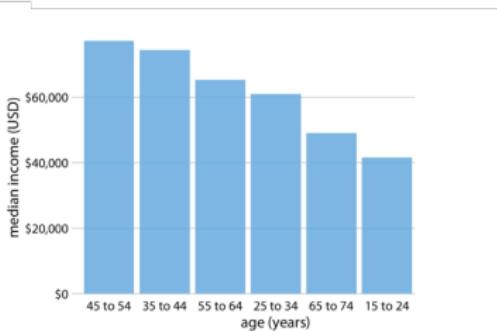
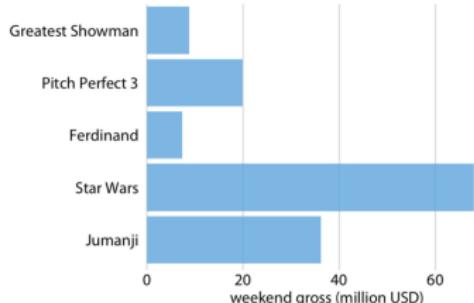


จาก barplot และ dot plot ที่ทำให้ดูเป็นตัวอย่าง ท่านมีข้อสังเกตใดบ้างหรือไม่?

# Order is matter



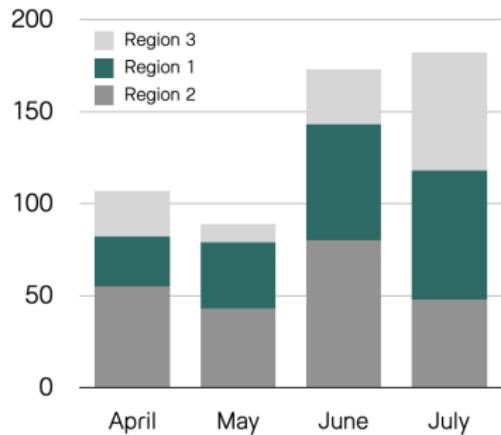
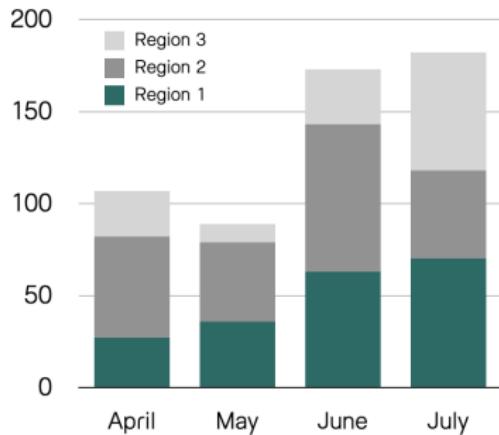
# Order is matter



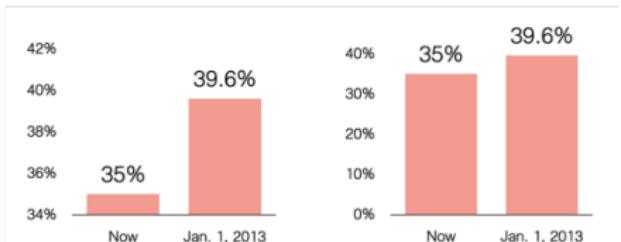
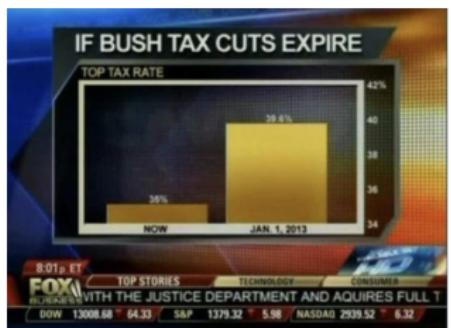
(Wilke, 2019)

# Order is matter

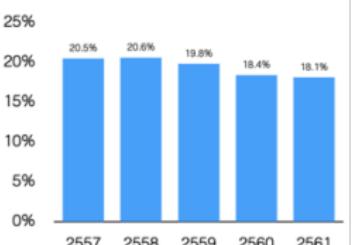
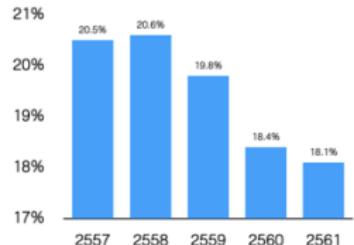
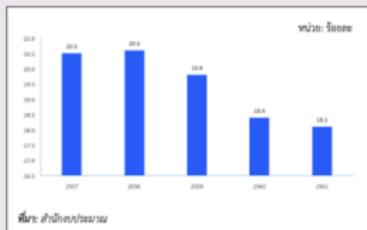
การเปรียบเทียบความแตกต่างระหว่าง category ที่ไม่ได้เริ่มจากเลน  $y=0$  ทำได้ยาก



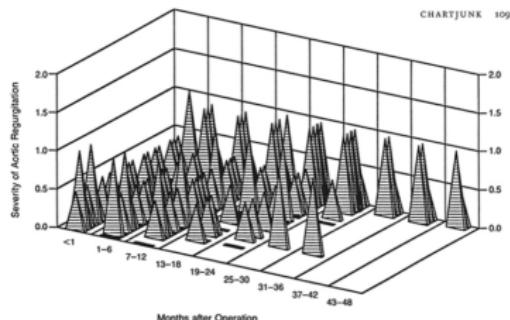
## Length of the bar/dot plot represents values



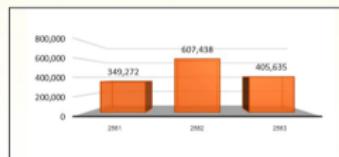
แผนภาพที่ 4.2 ร้อยละของงบประมาณทางการศึกษาต่องบประมาณ  
แผ่นดิน (ปีงบประมาณ พ.ศ. 2557-2561)



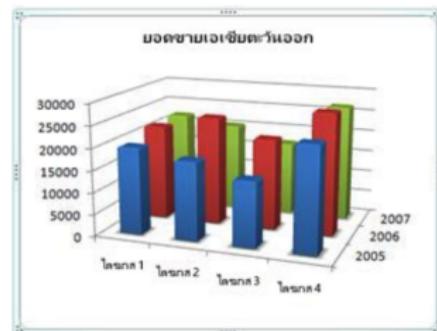
# Length of the bar/dot plot represents values



แผนภาพที่ 4.5 จำนวนนักเรียนพิการเรียนร่วมในโรงเรียนปกติที่สังกัดสำนักงานคณะกรรมการศึกษาขั้นพื้นฐาน (ปีการศึกษา 2561-2563)



ที่มา: สำนักงานคณะกรรมการศึกษาธิการแห่งชาติพื้นฐาน สำนักงานคณะกรรมการการศึกษาขั้นพื้นฐาน





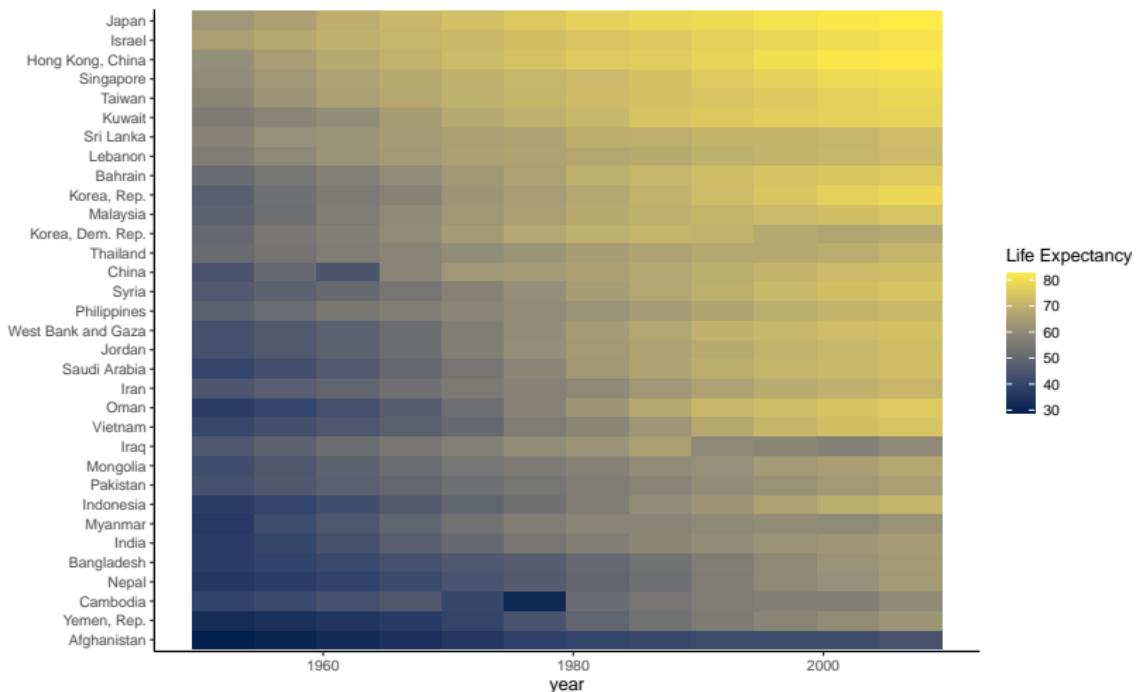
## Creating Heatmap using geom\_tile()

```
1 gapminder %>% filter(continent == "Asia") %>%
2   ggplot(aes(x = year,
3               y = reorder(country, lifeExp),
4               fill = lifeExp))+  

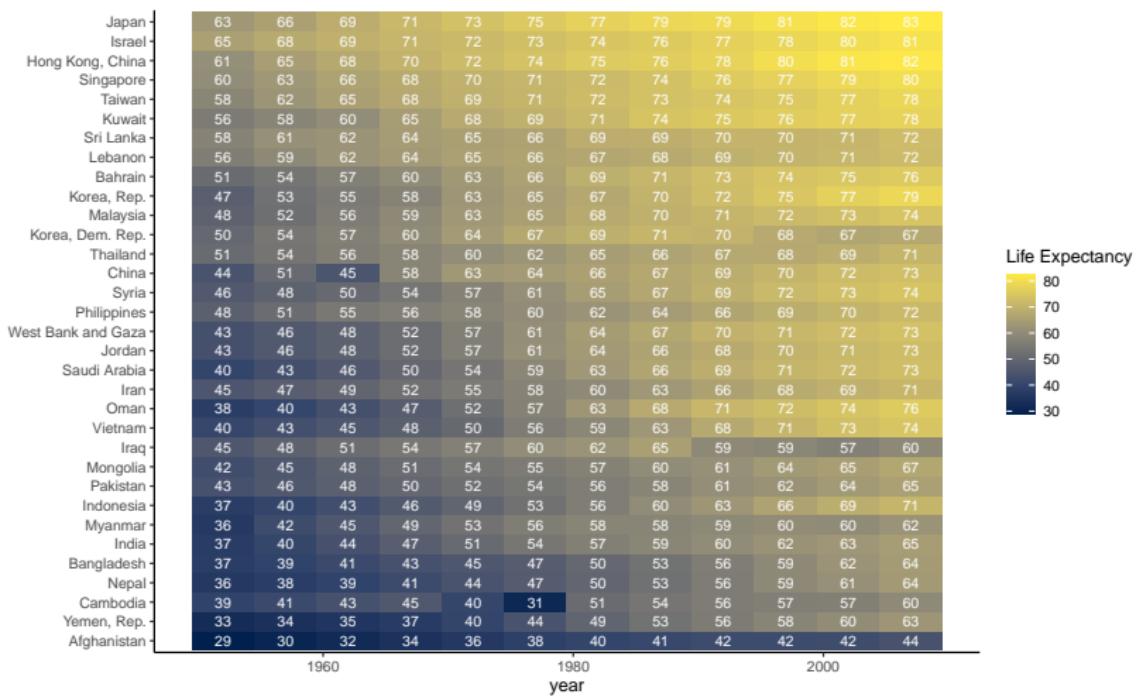
5   geom_tile()+
6   scale_fill_gradient2(low = "#A61F69",
7                         mid = "#F2921D",
8                         high ="#F1DBBF",
9                         midpoint = 65)+  

10  theme_classic()
```

## Creating Heatmap using `geom_tile()`

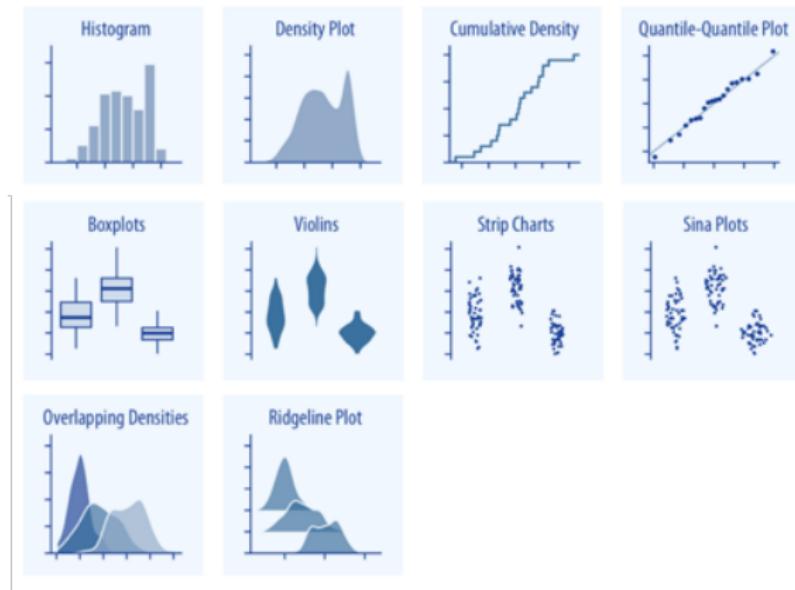


# Creating Heatmap using geom\_tile()



## Visualizing Distribution

# Visualizing Distribution

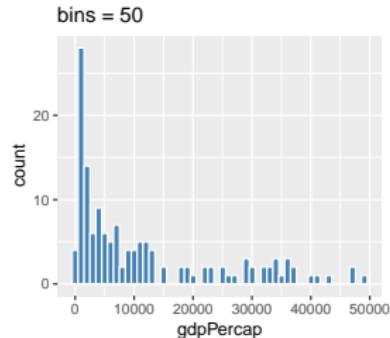
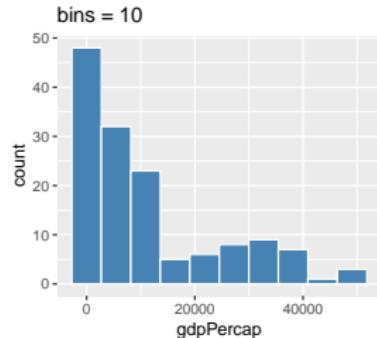
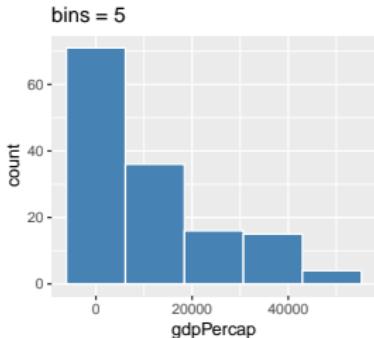


ที่มา : Wike (2019)

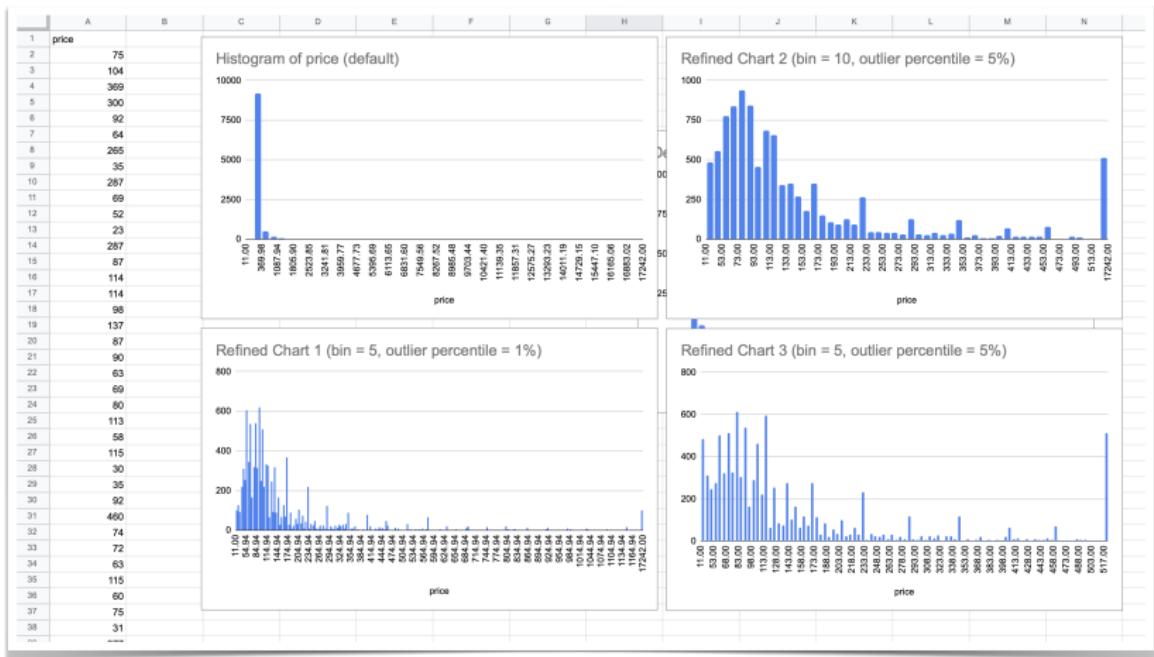
## Single Variable: Histogram

การสร้าง histogram และ density plot สามารถทำได้ด้วยฟังก์ชัน  
`geom_histogram()`

```
1 gapminder %>%
2   filter(year == "2007") %>%
3   ggplot(aes(x = gdpPercap))+
4   geom_histogram(fill = "steelblue", col="white")
```



# Histogram: binwidth



## Single Variable: Density plot

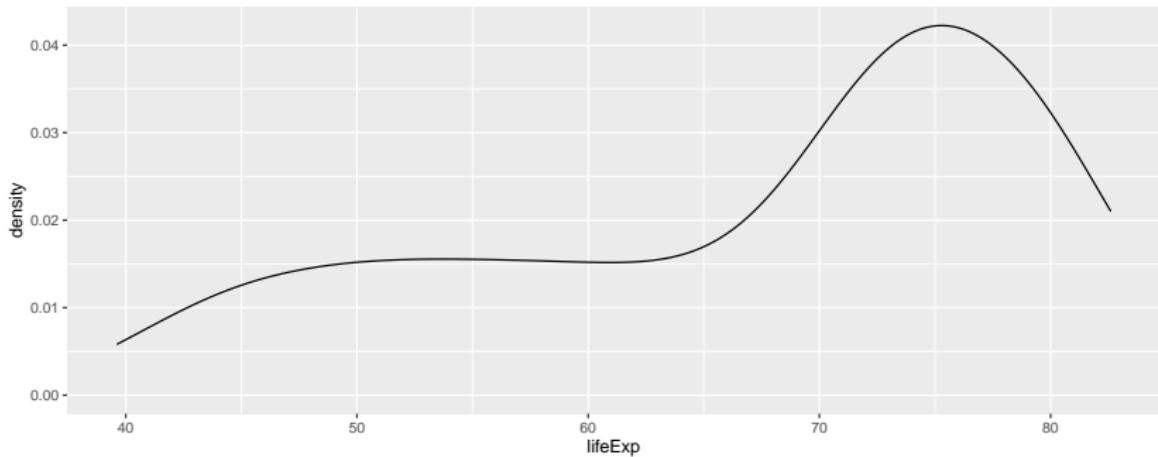
แผนภาพโค้งความหนาแน่น (density plot)

ใช้ประมาณการแจกแจงความน่าจะเป็นเบื้องหลังของตัวแปร โดยทั่วไปจะใช้อัลกอริทึมที่เรียกว่า kernel density estimation (KDE)

$$\hat{f}_h(x) = \frac{1}{n} \sum_{i=1}^n K_h(x - x_i)$$

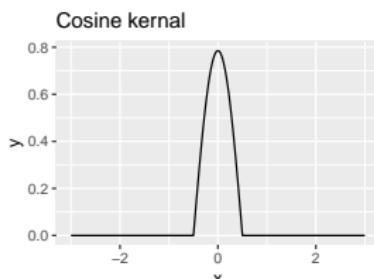
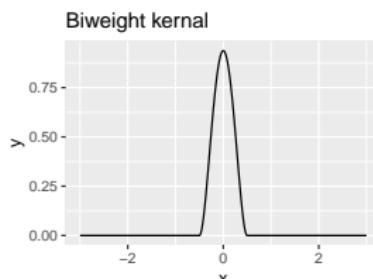
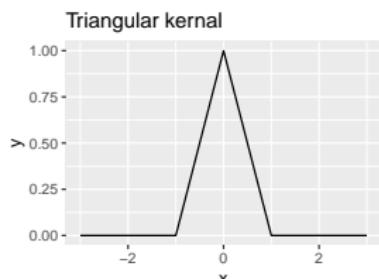
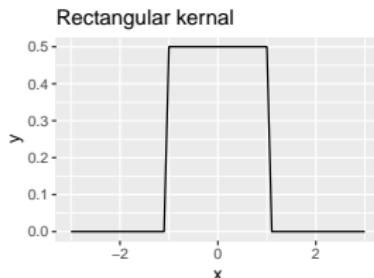
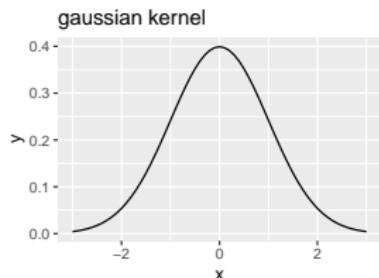
## Density plot

```
1 gapminder %>% filter(year == "2007") %>%
2   ggplot(aes(x = lifeExp))+
3   geom_density(kernel = "gaussian")
```



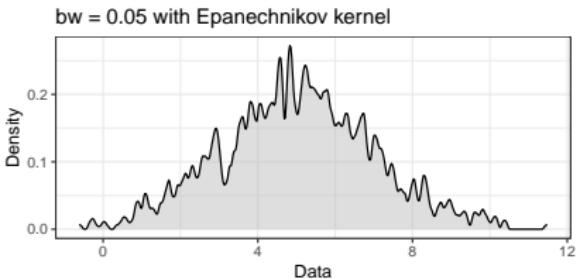
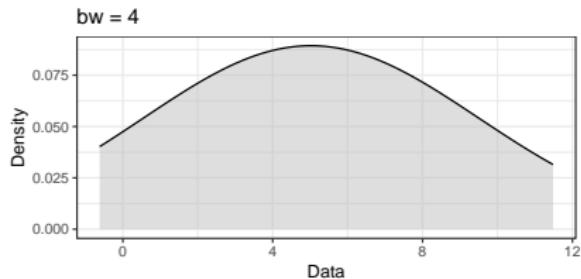
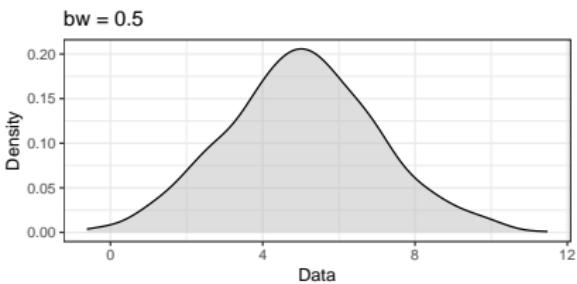
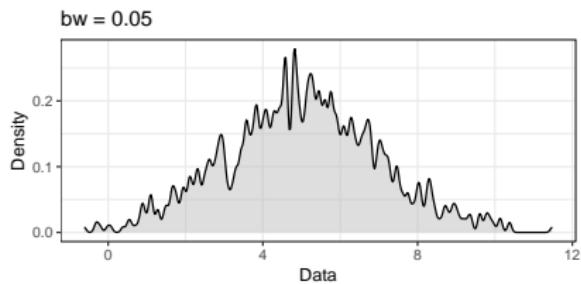
## Density plot: Kernel Functions

kernel function เป็นพารามิเตอร์ตัวหนึ่งของ density plot

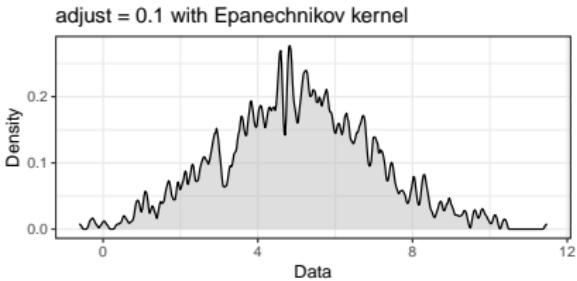
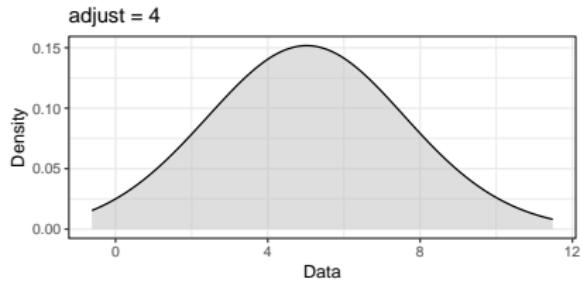
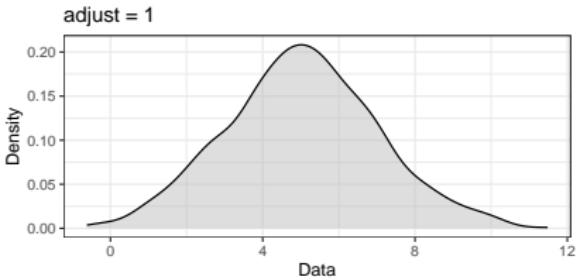
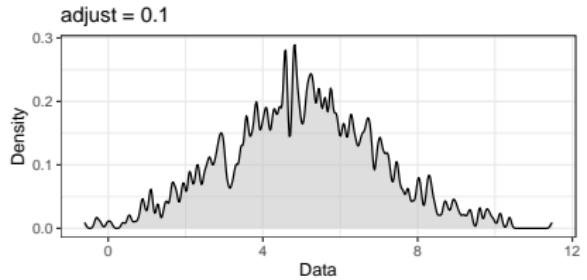


## Density plot: bandwidth

bandwidth เป็นพารามิเตอร์อีกด้านหนึ่งของ density plot ซึ่งสามารถกำหนดได้จาก อาร์กิวเม้นท์ bw หรือ adjust



## Density plot: bandwidth

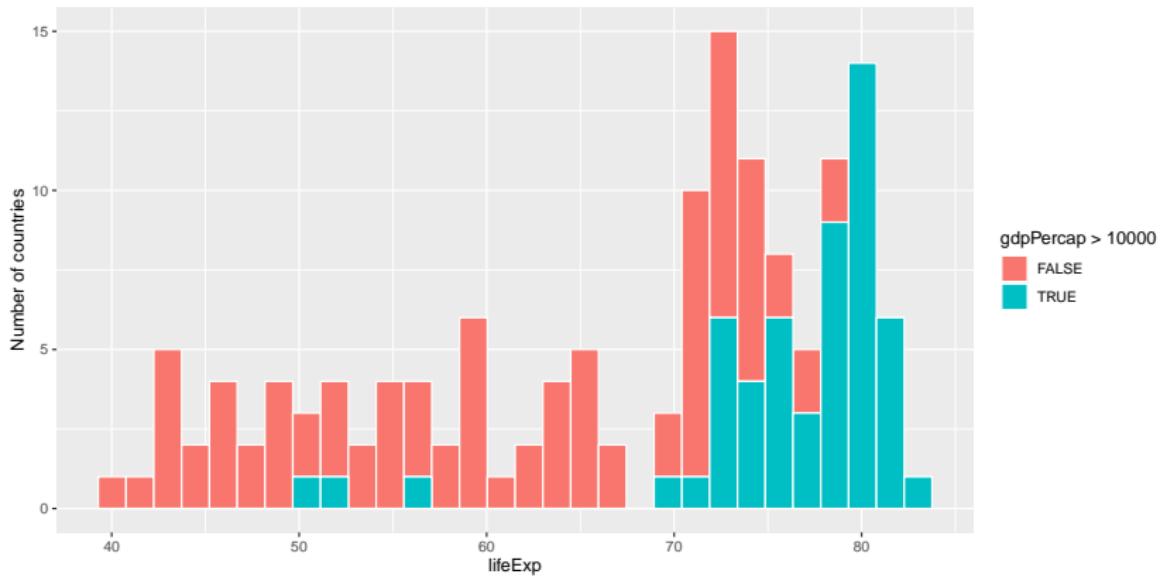


## Multiple Distribution: histogram

```
1 gapminder %>%
2   filter(year == "2007") %>%
3   ggplot(aes(x = lifeExp, fill = gdpPercap > 10000))+
4   geom_histogram(col="white")+
5   ylab("Number of countries")
```

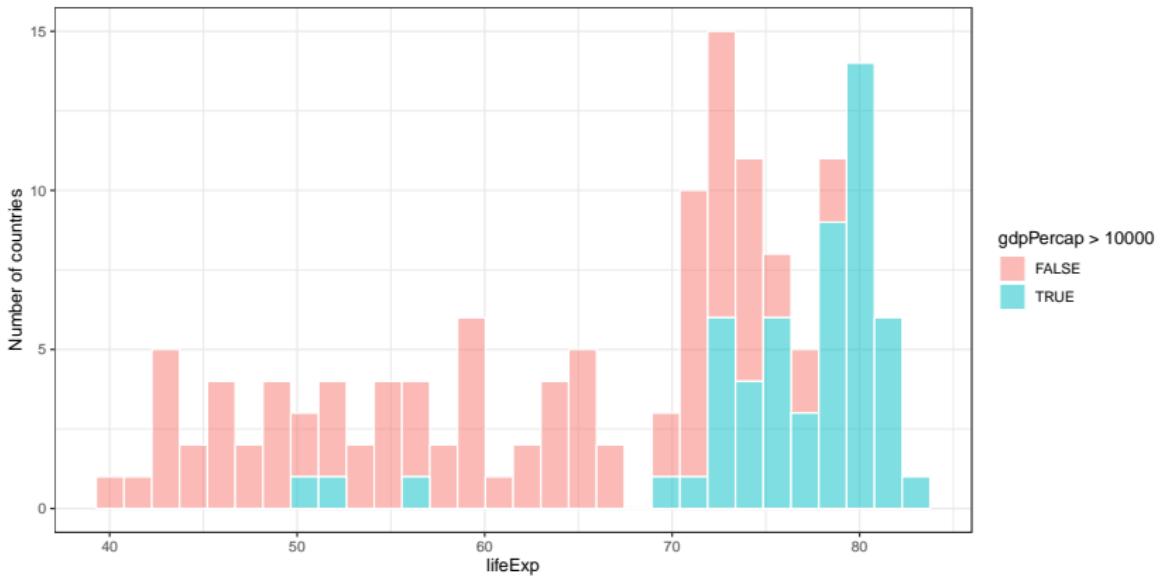
# Multiple Distribution: histogram

Good visualization ?



## Multiple Distribution: histogram

Good visualization ?



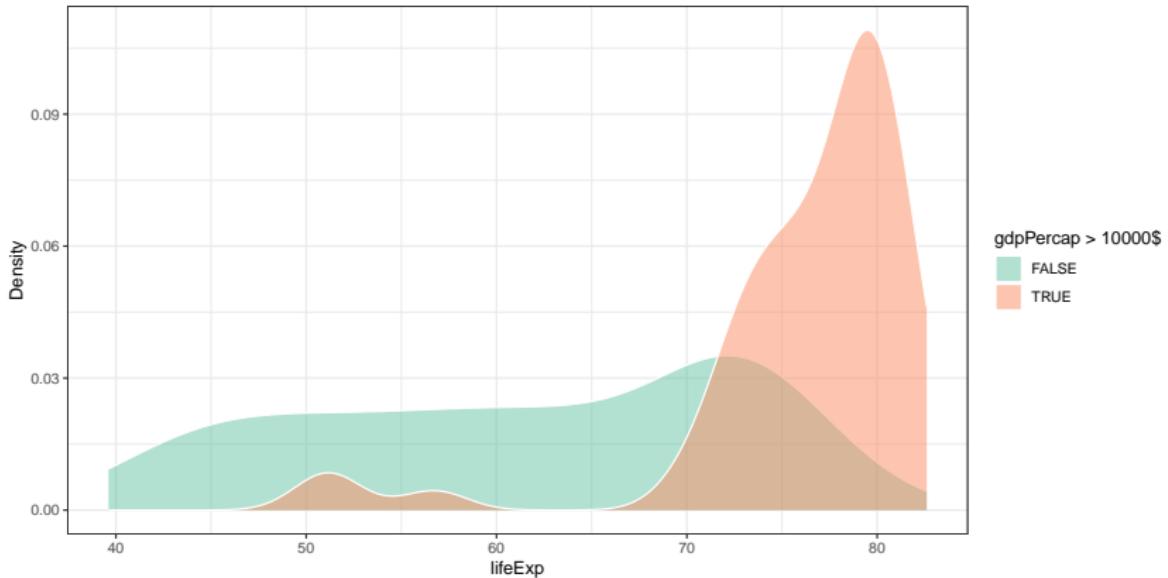
## Multiple Distribution: density plot

ลองสร้าง density plot ต่อไปนี้

```
1 gapminder %>%
2   filter(year == "2007") %>%
3   ggplot(aes(x = lifeExp, fill = gdpPercap > 10000)) +
4   geom_density(col="white", alpha = 0.5) +
5   ylab("Density") +
6   theme_bw() +
7   scale_fill_brewer(name= "gdpPercap > 10000$",
8                     type="qual",
9                     palette = "Set2")
```

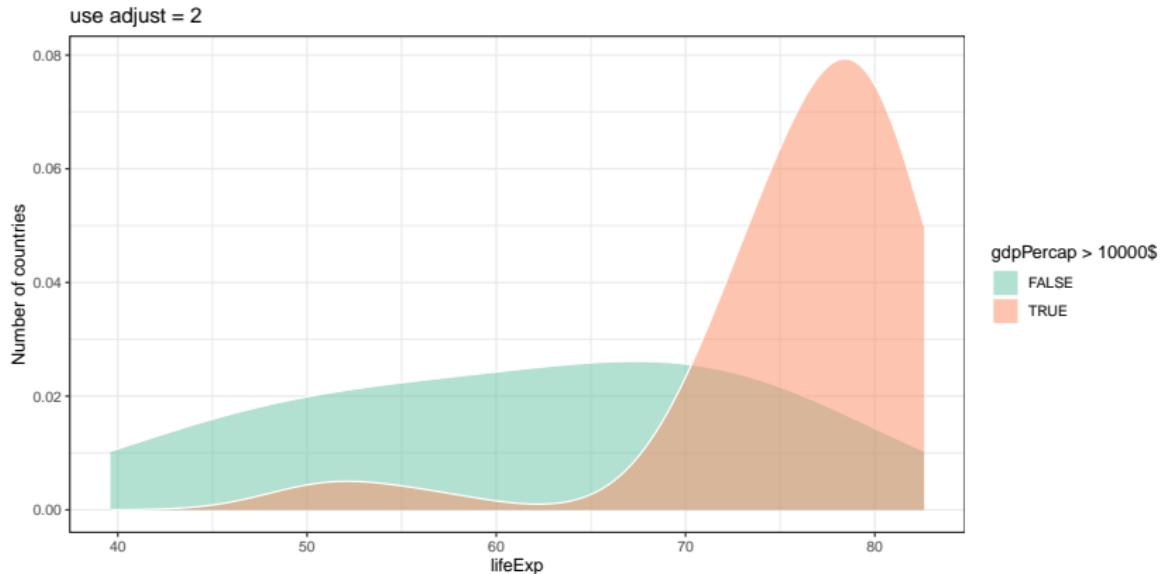
## Multiple Distribution: density plot

Better ? Good visualization ?



## Multiple Distribution: density plot

Better ? Good visualization ?



## Multiple Distribution: density plot

ขอให้ผู้เรียนสร้าง density plot

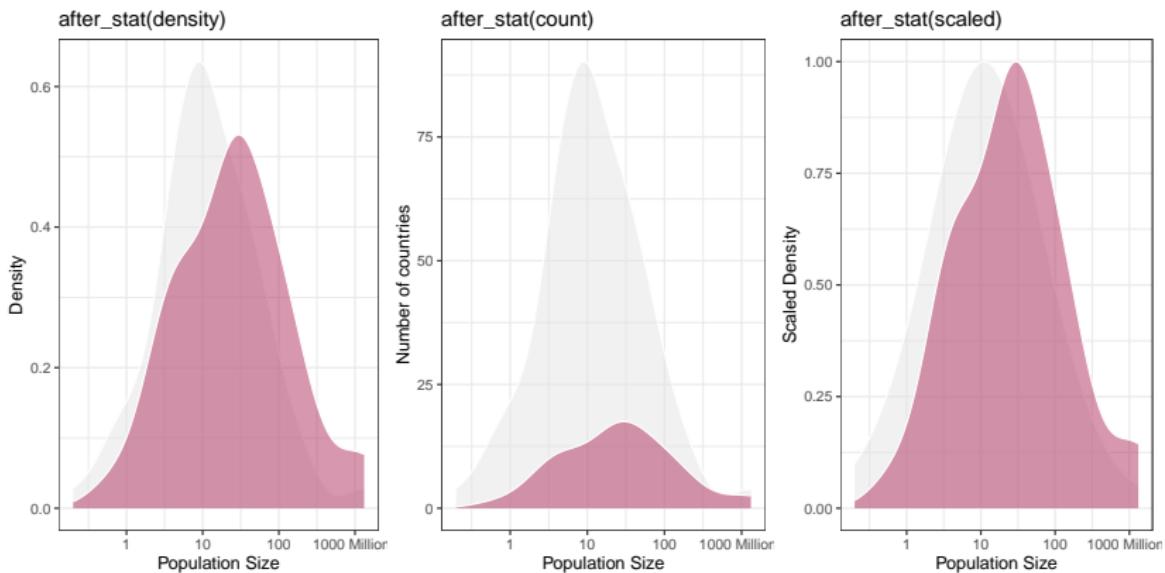
เพื่อเปรียบเทียบจำนวนประชากรของประเทศในกลุ่มเอเชีย กับทุกประเทศบนโลก

## Density plot: Statistical transformations

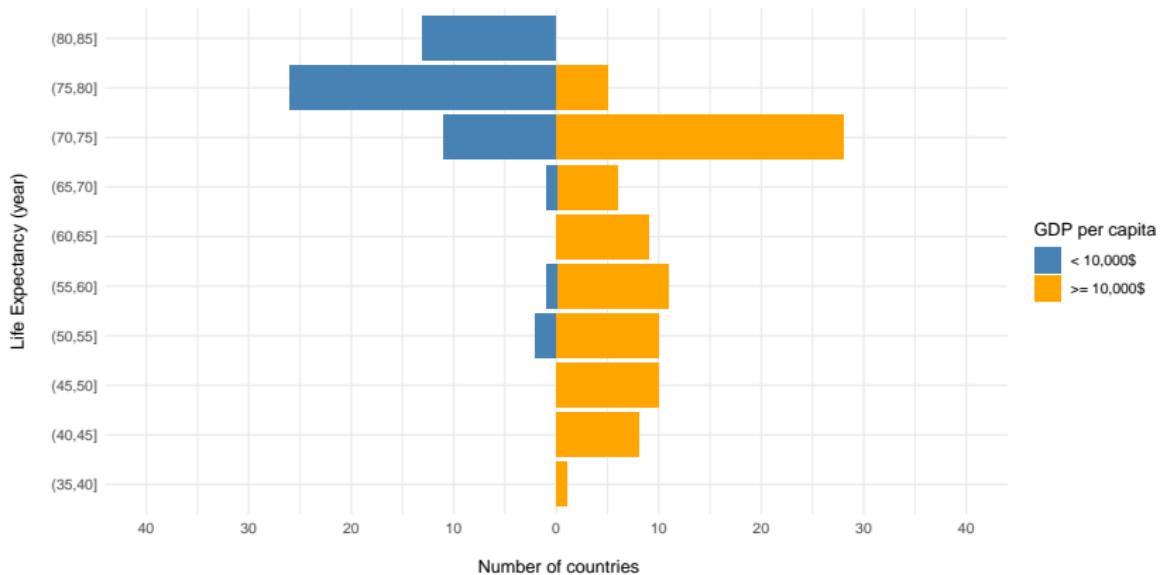
- ▶ `after_stat(density)`
- ▶ `after_stat(count)`
- ▶ `after_stat(scaled)`

## Density plot: Statistical transformations

จำนวนประชากรของประเทศในทวีปเอเชียเปรียบเทียบกับประเทศทั้งหมดบนโลก



## Multiple Distribution: pyramid chart



# Multiple Distribution: Pyramid chart

## Data preprocessing step

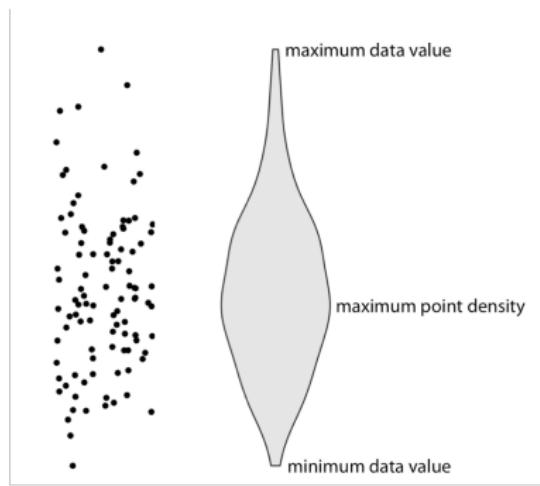
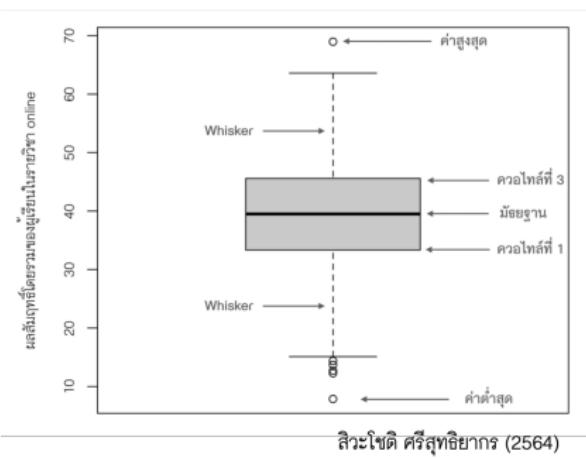
```
1 dat <- gapminder %>% filter(year == "2007")
2 dat$interval <- cut(dat$lifeExp, breaks=seq(25,85,5))
3 temp <- dat %>%
4   group_by(gdpPercap<10000, interval) %>%
5   count()
6
7 names(temp)[1]<-"gdp_select"
8 temp$gdp_select<-factor(temp$gdp_select,
9                           label=c("< 10,000$", ">= 10,000$"))
10 temp1 <- temp%>%filter(gdp_select == ">= 10,000$")
11 temp2 <- temp%>%filter(gdp_select == "< 10,000$")
```

## Multiple Distribution: pyramid chart

Create pyramid chart step

```
1 ggplot(data = temp, aes(x = n,
2                               y = interval,
3                               fill = gdp_select))+  
4   geom_bar(data = temp1, stat = "identity")+
5   geom_bar(data = temp2, stat = "identity")+
6   ylab("Life Expectancy (year)\n")+
7   scale_fill_manual(name = "GDP per capita",
8                     values = c("steelblue","orange"))+
9   scale_x_continuous(name = "\n Number of countries",
10                      limits = c(-40,40),
11                      breaks = seq(-40,40,10),
12                      labels = c(seq(40,10,-10),
13                                seq(0,40,10)))
```

## Multiple Distribution: Boxplot, Jitter and Violin plot

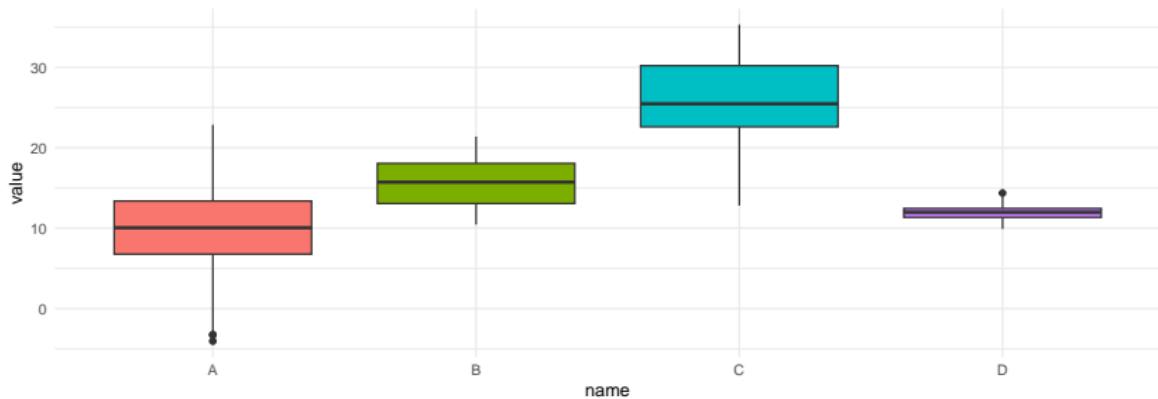


ข้อจำกัดของ boxplot คือสร้างจากค่าสถิติเพียง 5 ค่า ทำให้สูญเสีย information ไปพอสมควร ...

`geom_boxplot()`, `geom_jitter()`, and `geom_violin()`

## Boxplot ของ GdpPercap จำแนกตามทวีป

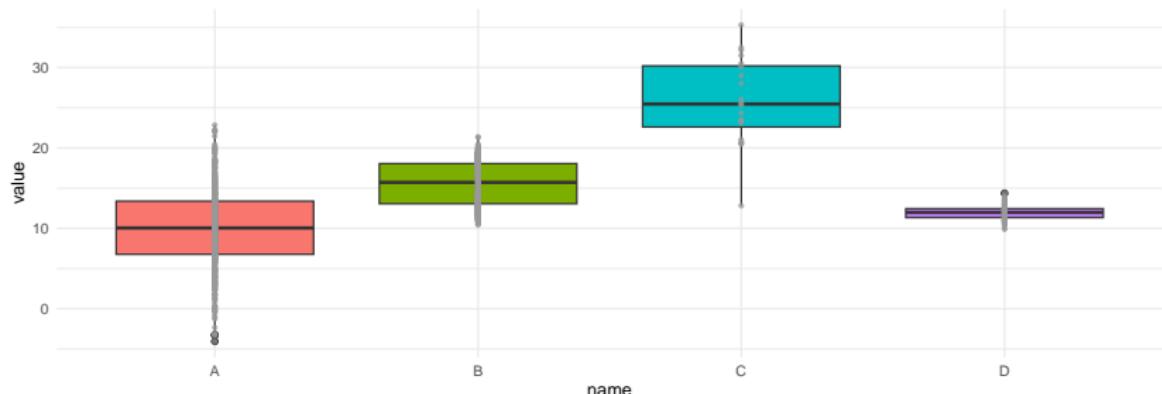
```
1 dat <- read.csv("boxplot.csv")
2 dat %>% ggplot(aes(x = name, y = value, fill = name))+ 
3   geom_boxplot(show.legend = F)+ 
4   theme_minimal()
```



## Adding Points

Better ? Good visualization ?

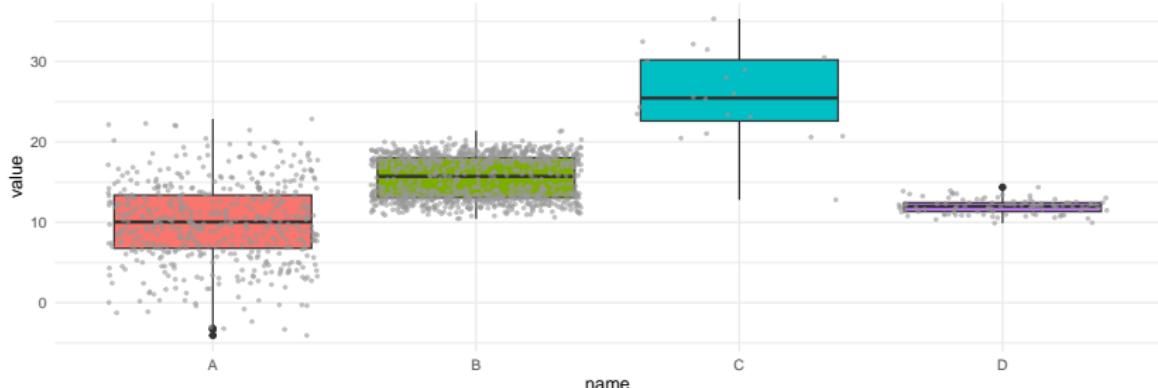
```
1 dat %>% ggplot(aes(x = name, y = value, fill = name))+  
2   geom_boxplot(show.legend = F)+  
3   geom_point(col = "grey60", alpha=0.7,  
4               size = 0.9, show.legend = F)+  
5   theme_minimal()
```



## Adding Jitter

Better ? Good visualization ?

```
1 dat %>% ggplot(aes(x = name, y = value, fill = name))+  
2   geom_boxplot(show.legend = F)+  
3   geom_jitter(col = "grey60", alpha=0.6,  
4                 size = 0.7, show.legend = F)+  
5   theme_minimal()
```



# Using Violin plot

Better ? Good visualization ?

```

1 dat %>% ggplot(aes(x = name, y = value, fill = name))+  

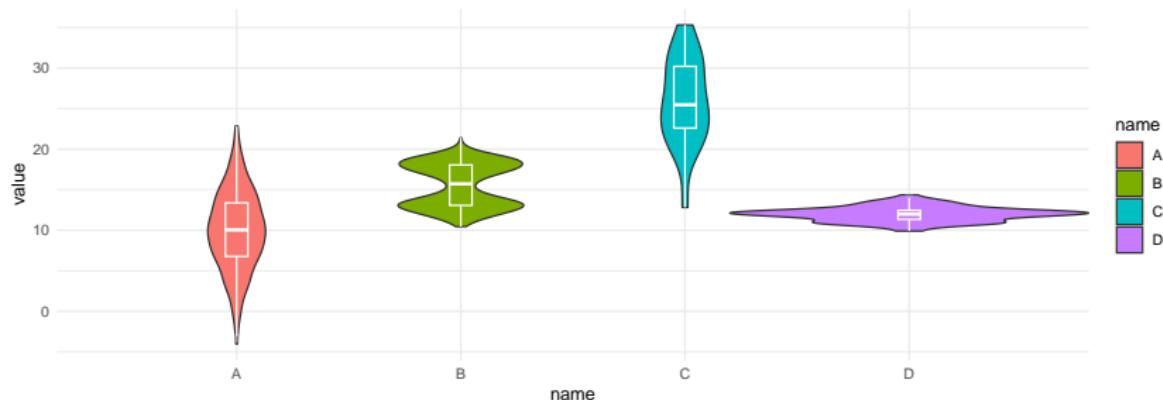
2   geom_violin(width = 1.6)+  

3   geom_boxplot(color = "white", show.legend = F,  

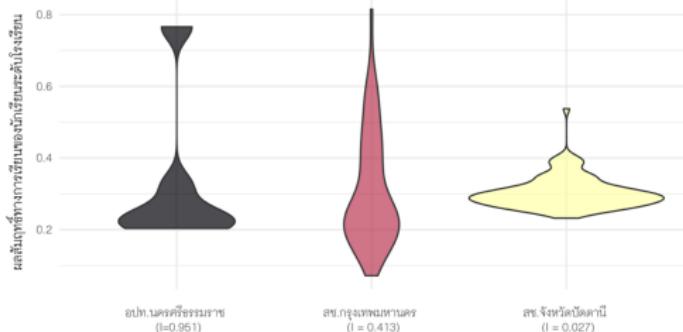
4                 width = 0.1, alpha = 0.2)+  

5   theme_minimal()

```



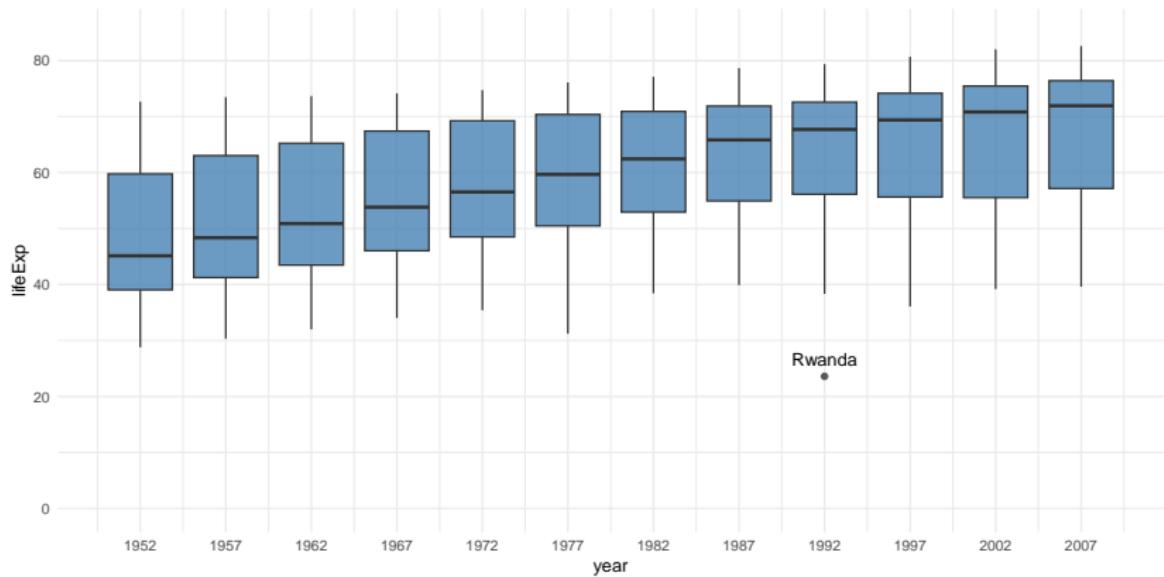
## Violin plot



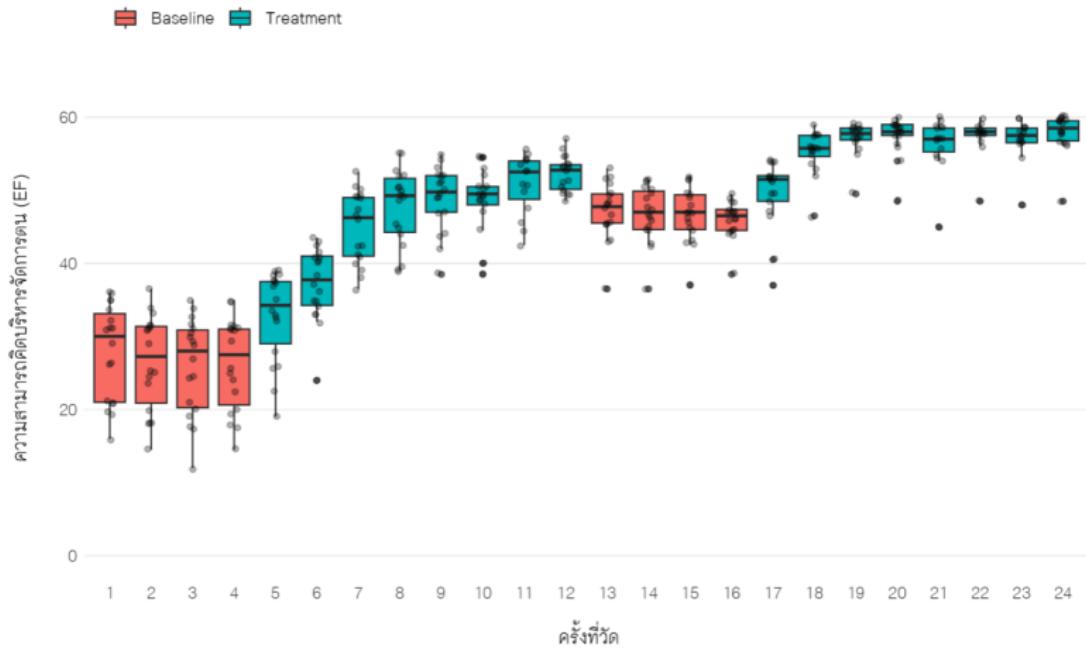
ภาพ 1 การแจกแจงของผลลัพธ์ทางการเรียนของนักเรียนระดับโรงเรียนภาษาในสังกัดที่มีความ  
เหลื่อมล้ำทางการศึกษาอยู่ในระดับสูงมาก ปานกลาง และต่ำ

สิริโชค ศรีสุทธิยก (2562)

# พัฒนาการด้านสุขภาพของประชากรโลก



# Time series Design



## Multiple Distribution: Ridge line plot

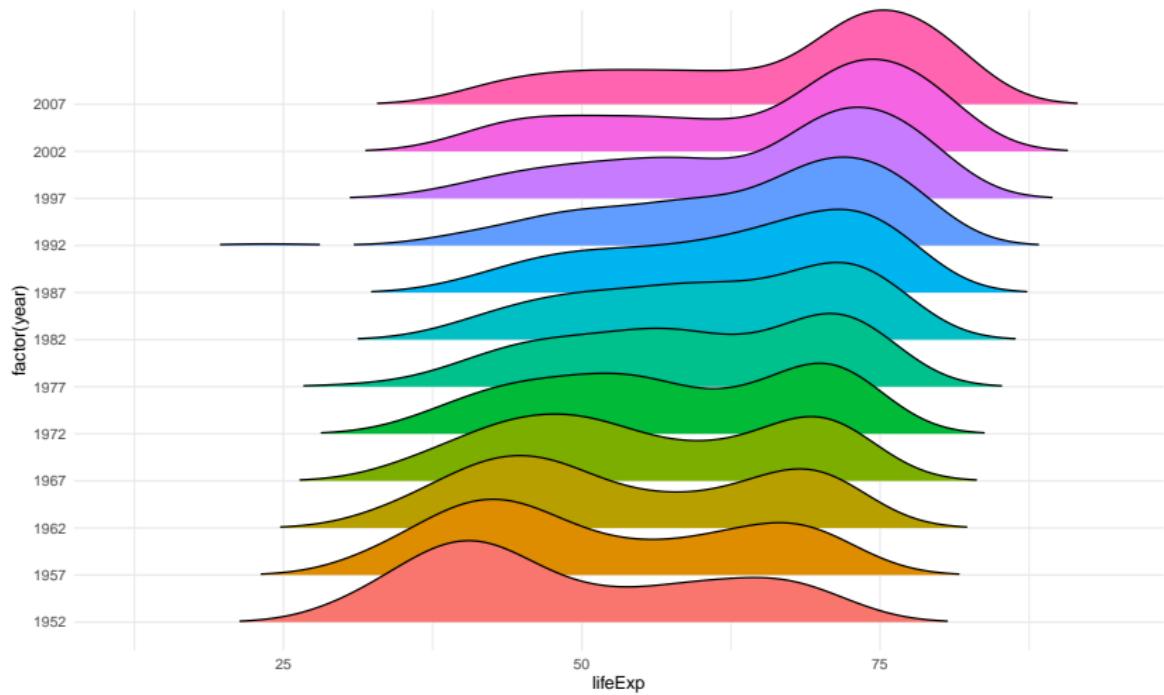
เหมาะสำหรับเปรียบเทียบการแจกแจงของตัวแปรที่มีการเปลี่ยนแปลงตามเวลา แต่ก็สามารถใช้เปรียบเทียบระหว่างกลุ่มหลายกลุ่มได้ package-ggridges เป็น ggplot2 extension ตัวหนึ่งที่สามารถใช้สร้างแผนภาพดังกล่าวได้

- ▶ `geom_ridgeline()`
- ▶ `geom_ridgeline_gradient()`
- ▶ `geom_density_ridges()`
- ▶ `geom_density_ridges_gradient()`

## Multiple Distribution: Ridge line plot

```
1 #install.packages("ggridges")
2 library(ggridges)
3 gapminder %>%
4   ggplot(aes(y = factor(year), x = lifeExp,
5             fill = factor(year)))+
6   geom_density_ridges(scale = 2,
7                       rel_min_height = 0.01,
8                       bandwidth = 4) +
9   scale_fill_discrete(guide = FALSE) +
10  theme_minimal()
```

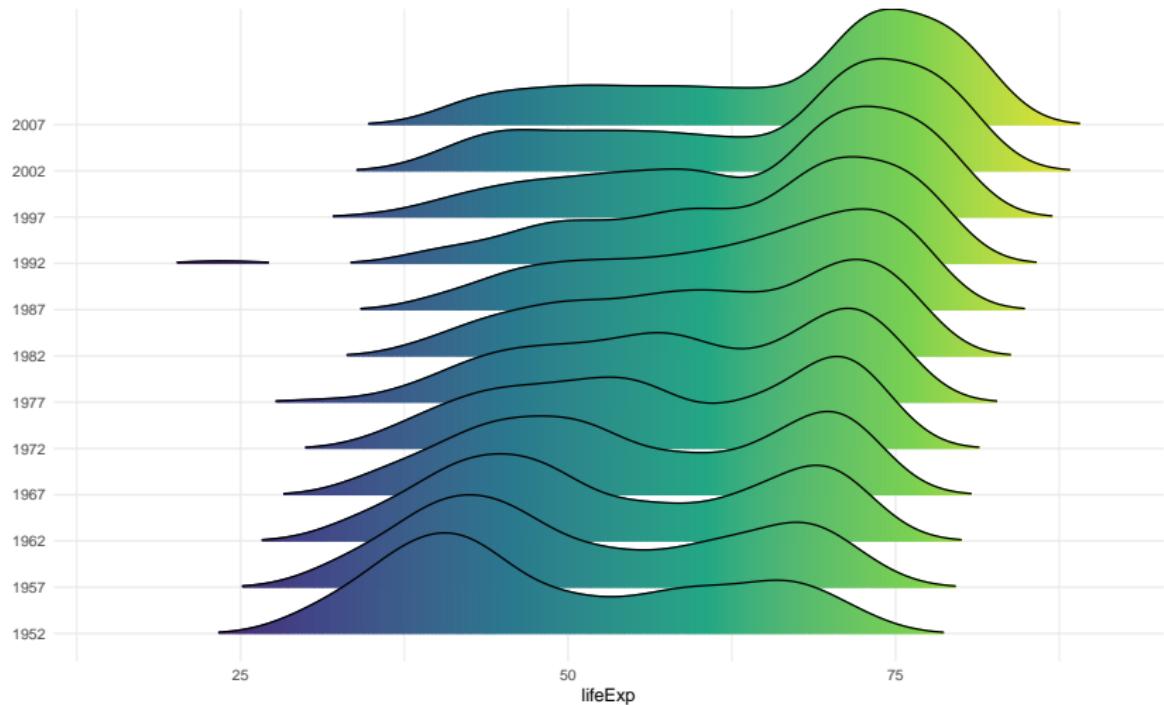
## Multiple Distribution: Ridge line plot



## Multiple Distribution: Ridge line plot

```
geom_density_ridges_gradient()  
  
1 gapminder %>%  
2   ggplot(aes(y = factor(year), x = lifeExp,  
3               fill = ...x...))+  
4     geom_density_ridges_gradient(scale = 2.5,  
5                                     rel_min_height = 0.01,  
6                                     bandwidth = 3) +  
7     ylab("")+  
8     scale_fill_viridis_c(option = "D") +  
9     guides(fill = F)+  
10    theme_minimal()
```

## Multiple Distribution: Ridge line plot



## Multiple Distribution: Ridge line plot

