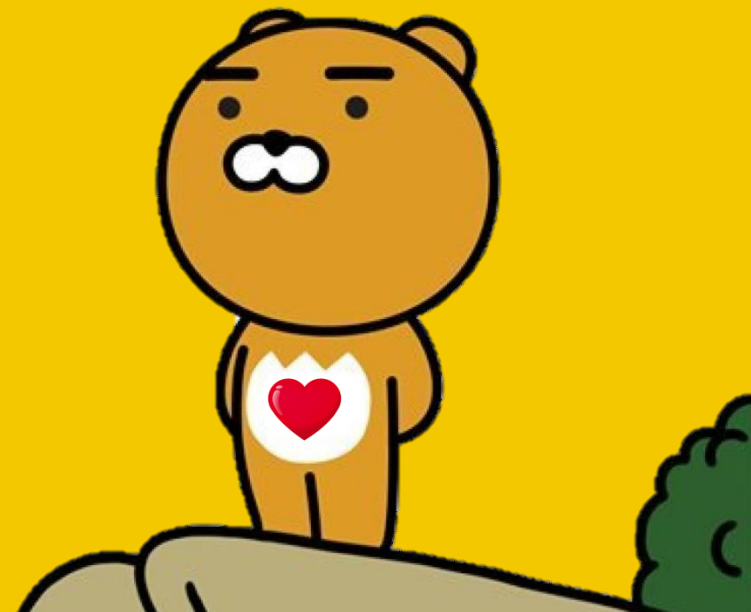
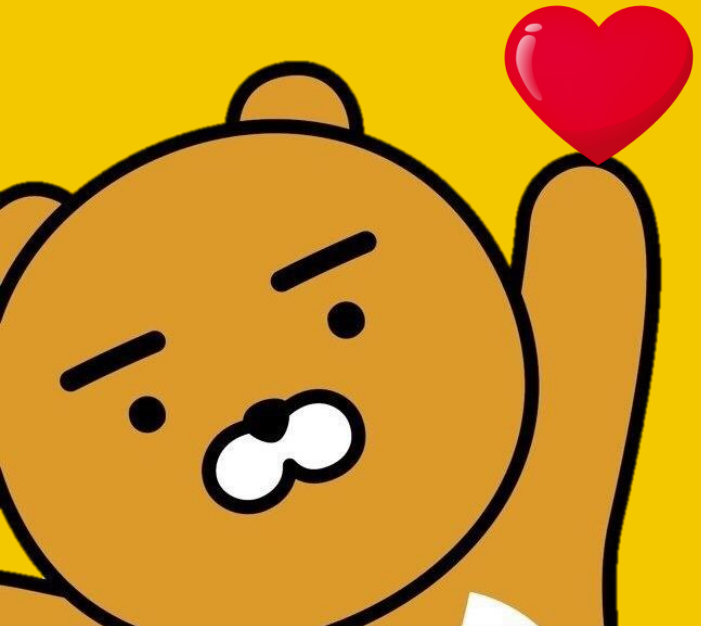
# R을 활용한 나라별 행복 점수 분석 및 시각화

발표자 : 김원섭



# CONTENTS



# 01. 데이터와 컬럼 설명

**Country : 나라명**

**Region : 지역**

**Happiness Rank : 행복 순위**

**Happiness Score : 행복 점수 (10점 만점)**

**Economy : GDP가 행복점수 계산에 관여하는 정도**

**Health : 평균 수명이 기여한 정도**

# 01. 데이터와 컬럼 설명

family : 가족이 행복 점수 계산에 기여하는 정도

freedom : 나라에서 자유가 행복 점수 계산에 기여하는 정도

Trust : 정부의 대한 신뢰감

Generosity : 관대함(친절한 정도)

Dystopia Residual : 디스토피아 나머지 점수

# 디스토피아 란?

디스토피아(Dystopia)는 세계에서 가장 행복한 사람이없는 상상의 나라입니다.

따라서 여섯 가지 핵심 변수에서 관찰 된 가장 낮은 점수는 디스토피아를 특징으로합니다.

세계 최저 소득, 최저 수명, 최저 관용, 부패, 자유 및 사회 지원이 가장 적은 나라에서는 인생이 매우 불행 할 것이므로 유토피아와 달리 "디스토피아(Dystopia)" 라고 합니다.

# 01. 데이터와 컬럼 설명

## 2015 행복지수 데이터

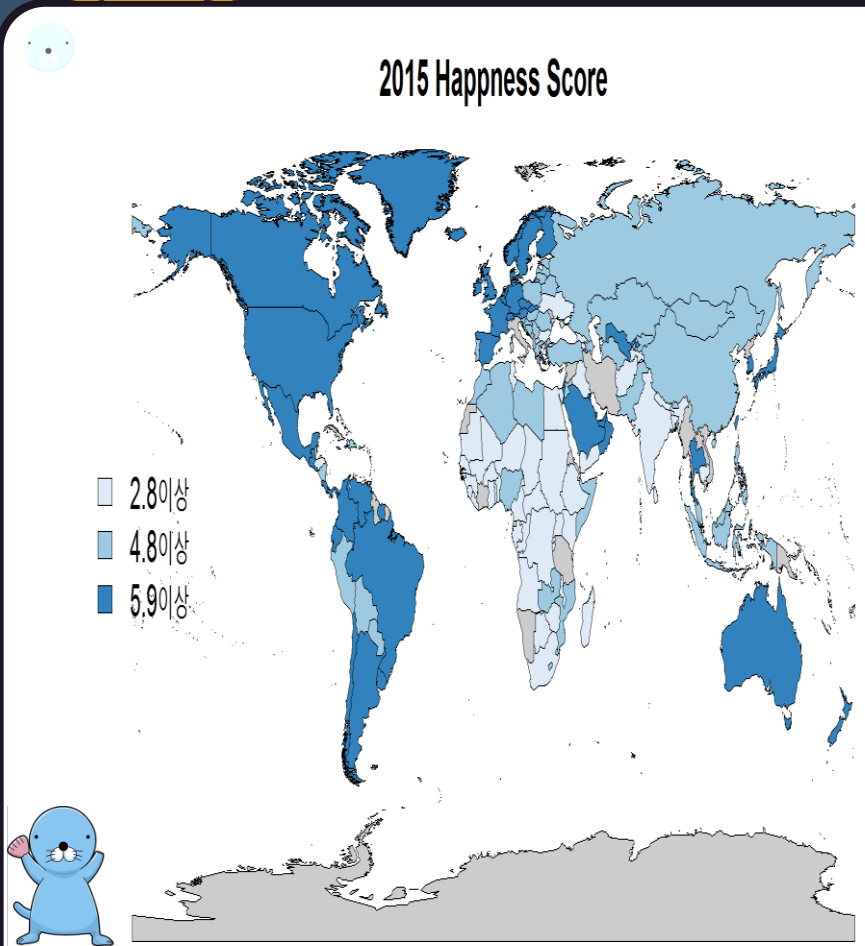
| | A | B | C | D | E | F | G | H | I | J | K | L |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 1 | Country | Region | Happiness_I | Happiness_S | Economy_G | Family | Health | Freedom | Trust | Generosity | Dystopia_Residual | |
| 2 | Switzerland | Western Eur | 1 | 7,587 | 1,39651 | 1,34951 | 0,94143 | 0,66557 | 0,41978 | 0,29678 | 2,51738 | |
| 3 | Iceland | Western Eur | 2 | 7,561 | 1,30232 | 1,40223 | 0,94784 | 0,62877 | 0,14145 | 0,4363 | 2,70201 | |
| 4 | Denmark | Western Eur | 3 | 7,527 | 1,32548 | 1,36058 | 0,87464 | 0,64938 | 0,48357 | 0,34139 | 2,49204 | |
| 5 | Norway | Western Eur | 4 | 7,522 | 1,459 | 1,33095 | 0,88521 | 0,66973 | 0,36503 | 0,34699 | 2,46531 | |
| 6 | Canada | North Americ | 5 | 7,427 | 1,32629 | 1,32261 | 0,90563 | 0,63297 | 0,32957 | 0,45811 | 2,45176 | |
| 7 | Finland | Western Eur | 6 | 7,406 | 1,29025 | 1,31826 | 0,88911 | 0,64169 | 0,41372 | 0,23351 | 2,61955 | |
| 8 | Netherlands | Western Eur | 7 | 7,378 | 1,32944 | 1,28017 | 0,89284 | 0,61576 | 0,31814 | 0,4761 | 2,4657 | |
| 9 | Sweden | Western Eur | 8 | 7,364 | 1,33171 | 1,28907 | 0,91087 | 0,6598 | 0,43844 | 0,36262 | 2,37119 | |
| 10 | New Zealan | Australia an | 9 | 7,286 | 1,25018 | 1,31967 | 0,90837 | 0,63938 | 0,42922 | 0,47501 | 2,26425 | |

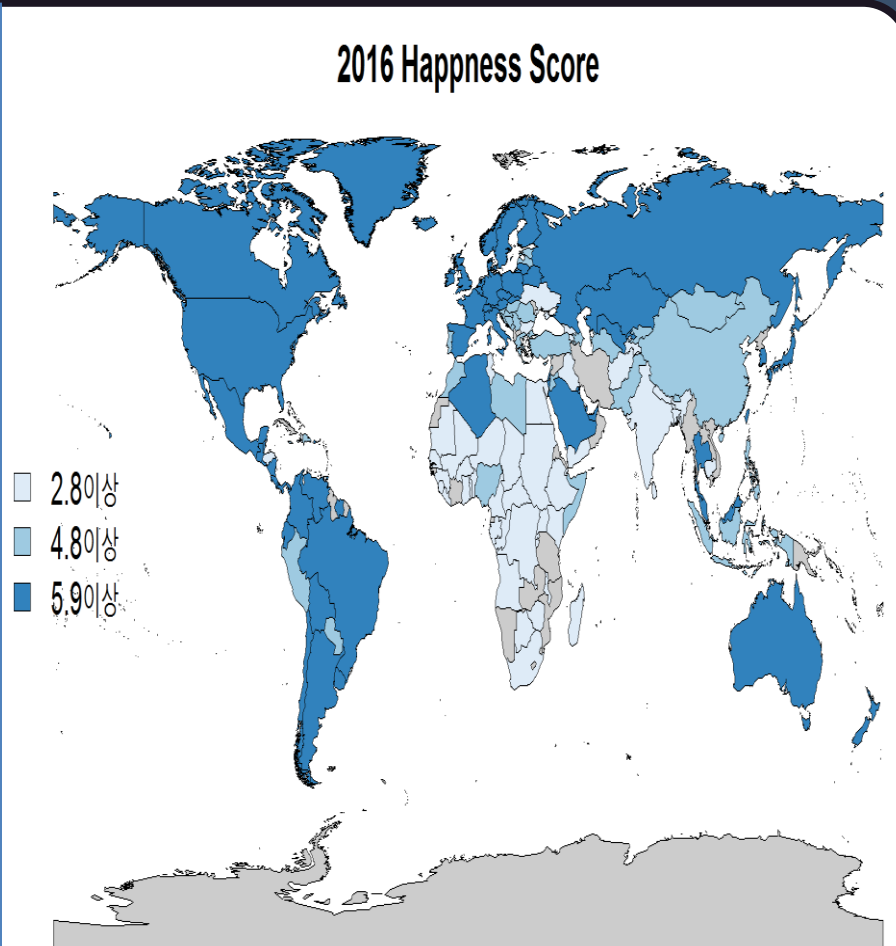
# 01. 데이터와 컬럼 설명

## 2016 행복지수 데이터

| | A | B | C | D | E | F | G | H | I | J | K | L |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 1 | Country | Region | Happiness F | Happiness_S | Economy_G | Family | Health | Freedom | Trust | Generosity | Dystopia_Residual | |
| 2 | Denmark | Western Eur | 1 | 7.526 | 1.44178 | 1.16374 | 0.79504 | 0.57941 | 0.44453 | 0.36171 | 2.73939 | |
| 3 | Switzerland | Western Eur | 2 | 7.509 | 1.52733 | 1.14524 | 0.86303 | 0.58557 | 0.41203 | 0.28083 | 2.69463 | |
| 4 | Iceland | Western Eur | 3 | 7.501 | 1.42666 | 1.18326 | 0.86733 | 0.56624 | 0.14975 | 0.47678 | 2.83137 | |
| 5 | Norway | Western Eur | 4 | 7.498 | 1.57744 | 1.1269 | 0.79579 | 0.59609 | 0.35776 | 0.37895 | 2.66465 | |
| 6 | Finland | Western Eur | 5 | 7.413 | 1.40598 | 1.13464 | 0.81091 | 0.57104 | 0.41004 | 0.25492 | 2.82596 | |
| 7 | Canada | North Ameri | 6 | 7.404 | 1.44015 | 1.0961 | 0.8276 | 0.5737 | 0.31329 | 0.44834 | 2.70485 | |
| 8 | Netherlands | Western Eur | 7 | 7.339 | 1.46468 | 1.02912 | 0.81231 | 0.55211 | 0.29927 | 0.47416 | 2.70749 | |
| 9 | New Zealan | Australia an | 8 | 7.334 | 1.36066 | 1.17278 | 0.83096 | 0.58147 | 0.41904 | 0.49401 | 2.47553 | |
| 10 | Australia | Australia an | 9 | 7.313 | 1.44443 | 1.10476 | 0.8512 | 0.56837 | 0.32331 | 0.47407 | 2.5465 | |

# 02. 나라별 행복점수 시각화1(maptools)



# maptools 패키지

| | | |
|---|---|---|
| wrld_simpl | S4 [246 x 11] (sp::SpatialPolygons[ | S4 object of class SpatialPolygonsDataFrame |
| data | list [246 x 11] (S3: data.frame) | A data.frame with 246 rows and 11 columns |
| polygons | list [246] | List of length 246 |
| plotOrder | integer [246] | 145 175 24 209 30 21 ... |
| bbox | double [2 x 2] | -180.0 -90.0 180.0 83.6 |
| proj4string | S4 (sp::CRS) | S4 object of class CRS |

| | | |
|---|---|---|
| data | list [246 x 11] (S3: data.frame) | A data.frame with 246 rows and 11 columns |
| FIPS | factor | Factor with 246 levels: "AC", "AG", "AJ", "AL", "AM", "AO" , ... |
| ISO2 | factor | Factor with 246 levels: "AG", "DZ", "AZ", "AL", "AM", "AO" , ... |
| ISO3 | factor | Factor with 246 levels: "ATG", "DZA", "AZE", "ALB", "ARM", "AGO" , ... |
| UN | integer [246] | 28 12 31 8 51 24 ... |
| NAME | factor | Factor with 246 levels: "Antigua and Barbuda", "Algeria", "Azerbaijan", "Albania , ... |

# 02. 나라별 행복점수 시각화1(maptools 코드)

```
happy2015 <- read.csv("C:₩₩data₩₩happy2015_2.csv",header = T)

library(maptools)

happy_5.9 = wrld_simpl@data$NAME %in% c("나라이름", ... )
        aaa<-ifelse(happy_5.9==T,1,happy_5.9)

happy_4.8 = wrld_simpl@data$NAME %in% c("나라이름", ... )
        aaa<-ifelse(happy_4.8==T,2,aaa)

happy_2.8 = wrld_simpl@data$NAME %in% c("나라이름", ...)
        aaa<-ifelse(happy_2.8==T,3,aaa)
```

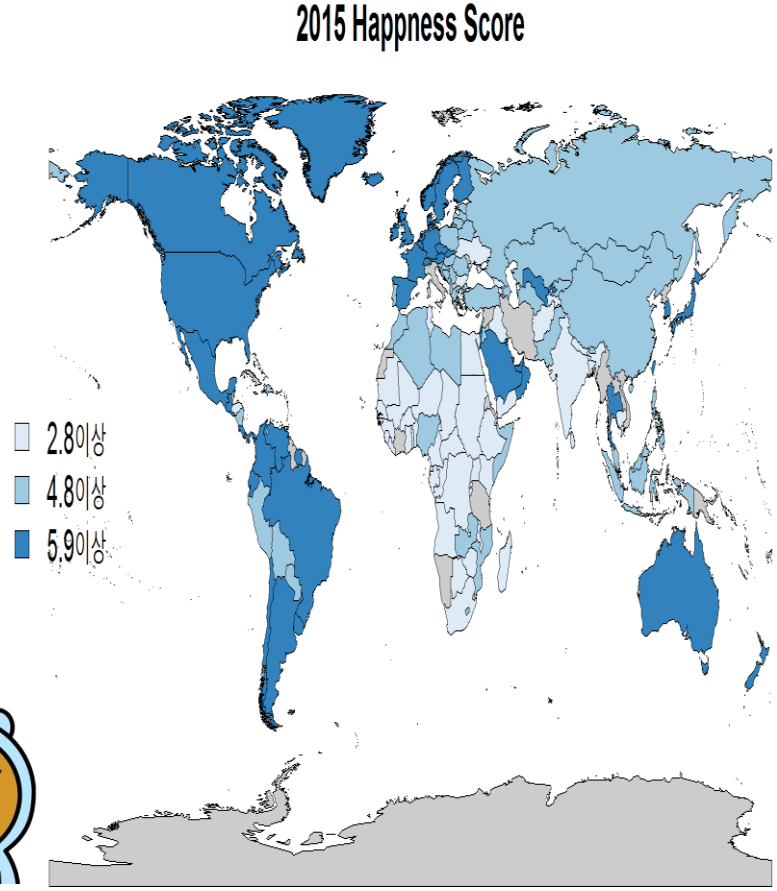
# 02. 나라별 행복점수 시각화1(maptools 코드)

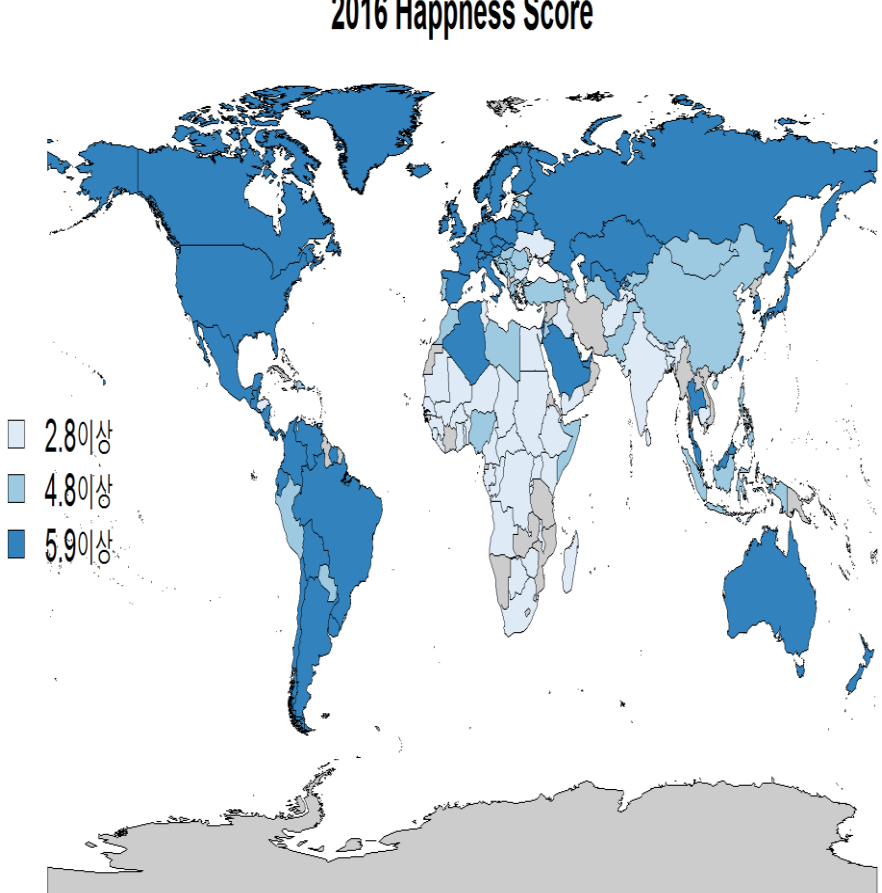
```
library(OIdata)
library(RColorBrewer)
library(classInt)

colors <- c("#00C853","#FFFF00","#FFAB00")
data(state)

nclr <- 3
min <- 0
max <- 100
breaks <- (max - min) / nclr

plotclr <- brewer.pal(nclr, "Blues")
plotvar <- state$coal
```

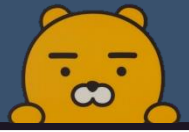
# 02. 나라별 행복점수 시각화1(maptools 코드)

```
class <- classIntervals(plotvar,
                        nclr,
                        style = "fixed",
                        fixedBreaks = seq(min, max, breaks))

colcode <- findColors(class, plotclr)

plot(wrld_simpl, col = c(gray(.80),"#3182BD","#9ECAE1","#DEEBF7")[aaa+1],
      main = "2015 Happness Score", cex.main=3.0)

legend("left",
          legend = c('2.8이상','4.8이상','5.9이상'),
          fill = attr(colcode, "palette"),
          cex = 2.7,
          bty = "n")
```

# 02. 나라별 행복점수 시각화1(maptools)

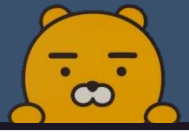




# dplyr 패키지 : chain operations %>%

**chain operation으로 굴비 꿰듯이 엮어서 물 흐르듯이 데이터 조작, 전처리를 할 수 있는 매끈한 문법을 생성**

# dplyr 패키지 : chain operations %>%

- 어떨 한 경우 chain operations 이 유용한가?

여러 단계의 절차를 필요로 해서 중간 결과를 저장 후 그 객체를 후속 절차에서 받아서 사용해야 하는 경우

ex) "Cars 데이터프레임에서 차생산국가, 차종, 실린더개수별로 차 가격과 고속도로연비의 평균을 구하되, 차가격 평균 10 초과 & 고속도로연비 25 초과하는 경우만 선별 할때

Cars 데이터프레임에서 %>% 제조생산국, 차종, 실린더개수별 %>% 차 가격과 고속도로 연비 변수에 평균 %>% 가격 평균은 10을 넘고 & 고속도로 연비는 25를 초과하는 경우

이런 방법으로 사용할 수 있습니다.

# 03. 나라별 행복점수 시각화 코드 (ggalt)

```
library(ggalt)
library(dplyr)
library(ggplot2)
library(readr)

xx15<-happy2015 %>% select(Country,Region,yy15=Happiness_Score)
xx16<-happy2016 %>% select(Country,Region,yy16=Happiness_Score)

score<-inner_join(xx15,xx16) %>% mutate(score_diff= yy16-yy15) %>%
        filter(score_diff>0)
```

# 03. 나라별 행복점수 시각화 코드 (ggalt)

```
score<-inner_join(xx15,xx16) %>% mutate(score_diff= yy16-yy15) %>%
        filter(score_diff>0)
```

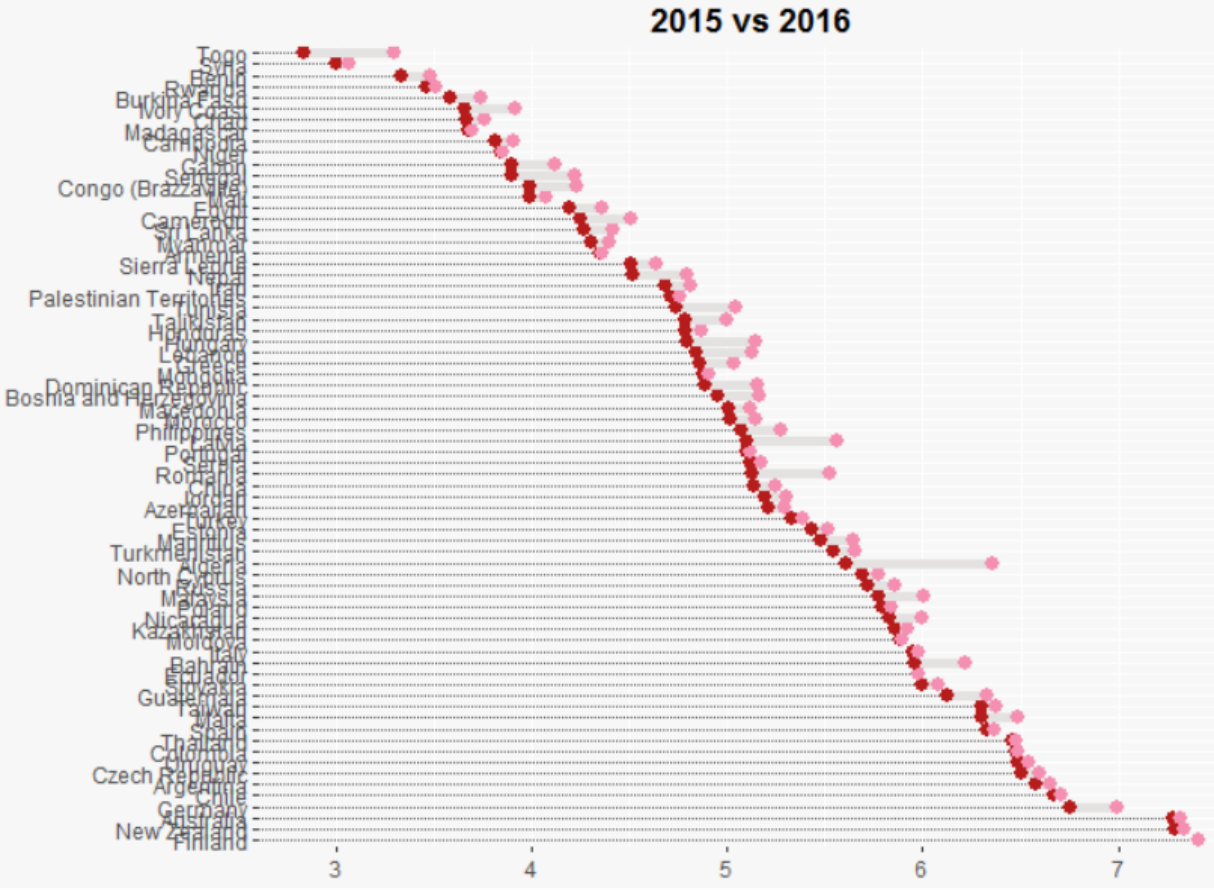
```
Joining, by = c("Country", "Region")
Warning message:
Column `Country` joining factors with different levels, coercing to character vector
```

```
score$Country <- factor(score$Country, levels=as.character(score$Country))
```

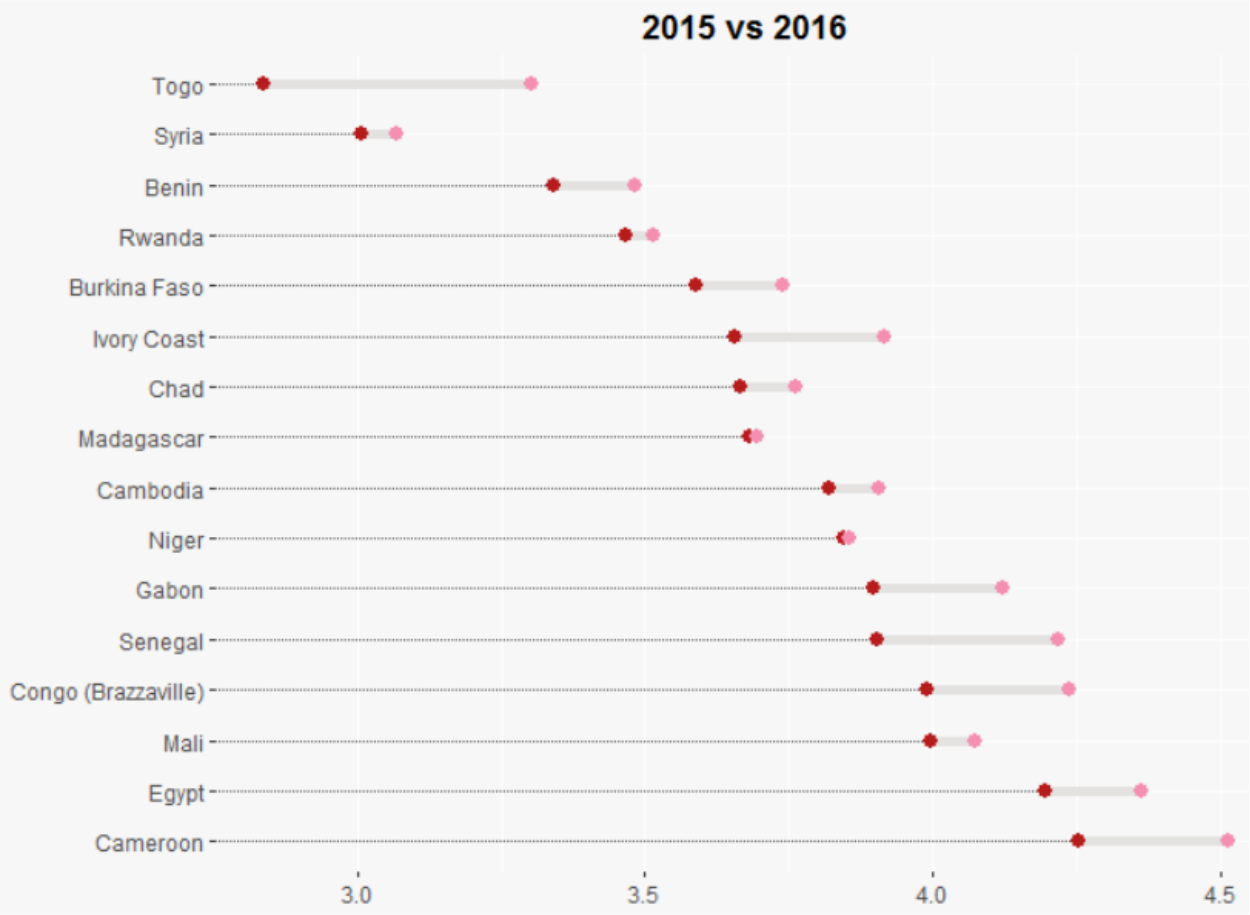
# 03. 나라별 행복점수 시각화 코드 (ggalt)

```
gg <- ggplot(score, aes(x=yy15, xend=yy16, y=Country, group=Country)) +
                geom_dumbbell(size=2, color="#e3e2e1",
                colour_x = "#B71C1C", colour_xend = "#F48FB1",
                dot_guide=TRUE, dot_guide_size=0.25) +
                labs(x=NULL, y=NULL, title="2015 vs 2016" ) +
                theme(plot.title = element_text(hjust=0.5, face="bold"),
                plot.background=element_rect(fill="#f7f7f7"),
                panel.background=element_rect(fill="#f7f7f7") )

plot(gg)
```

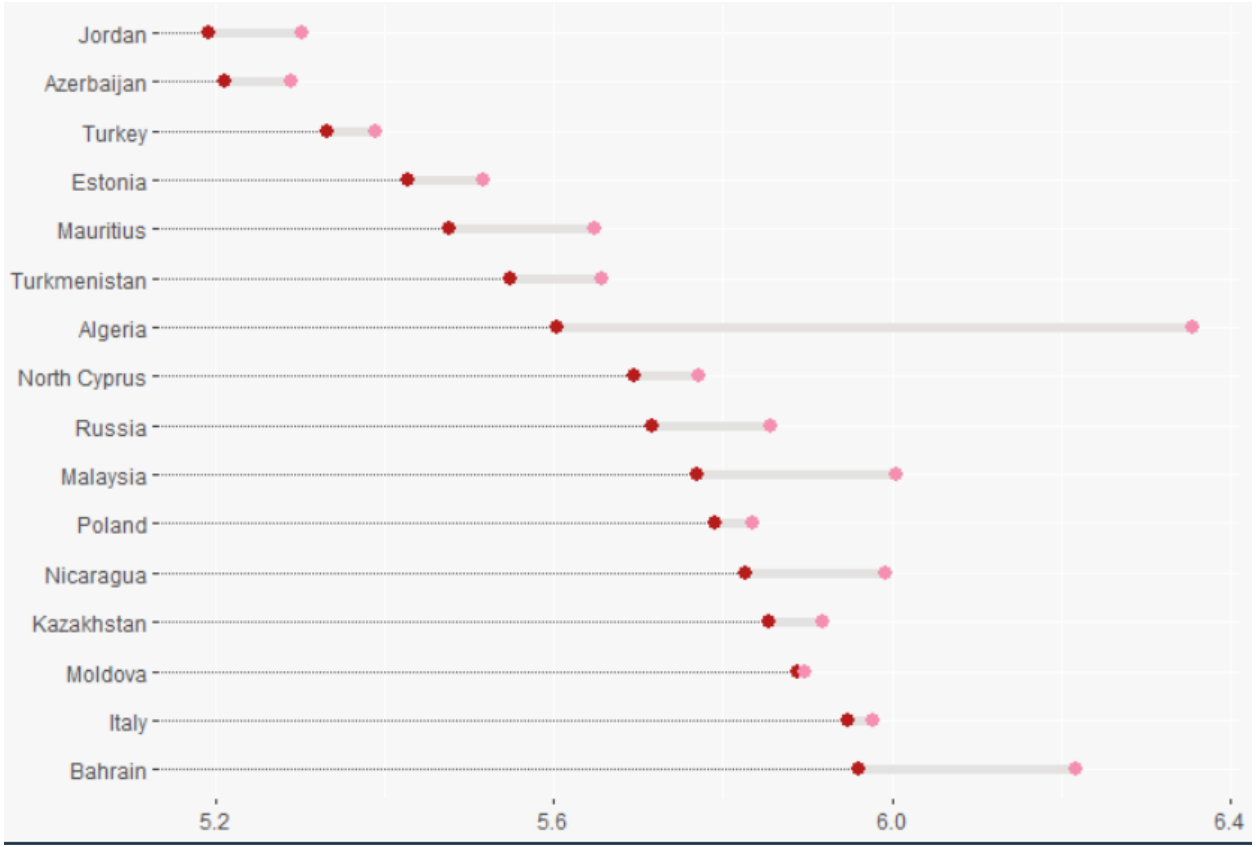
# 03. 나라별 행복점수 시각화 (ggalt)



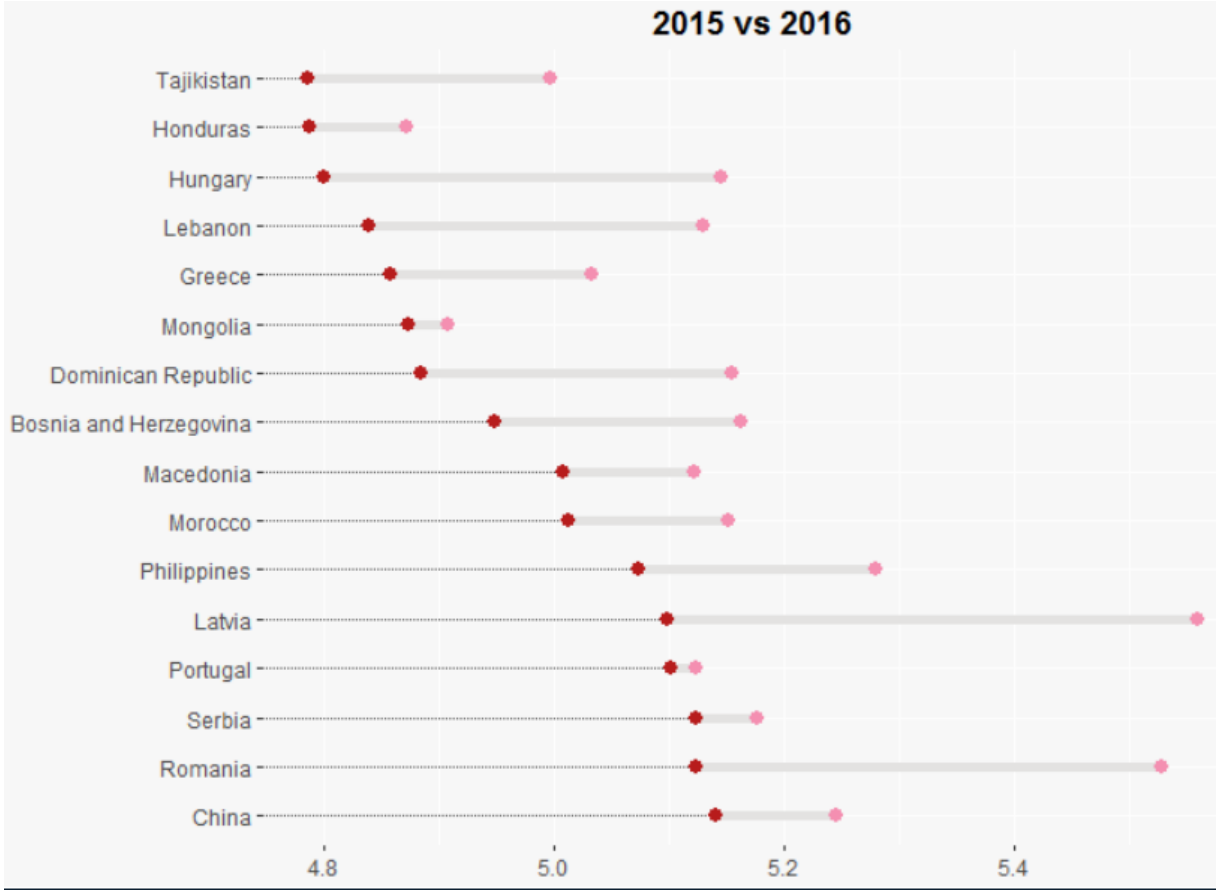
# 03. 나라별 행복점수 시각화 (ggalt)



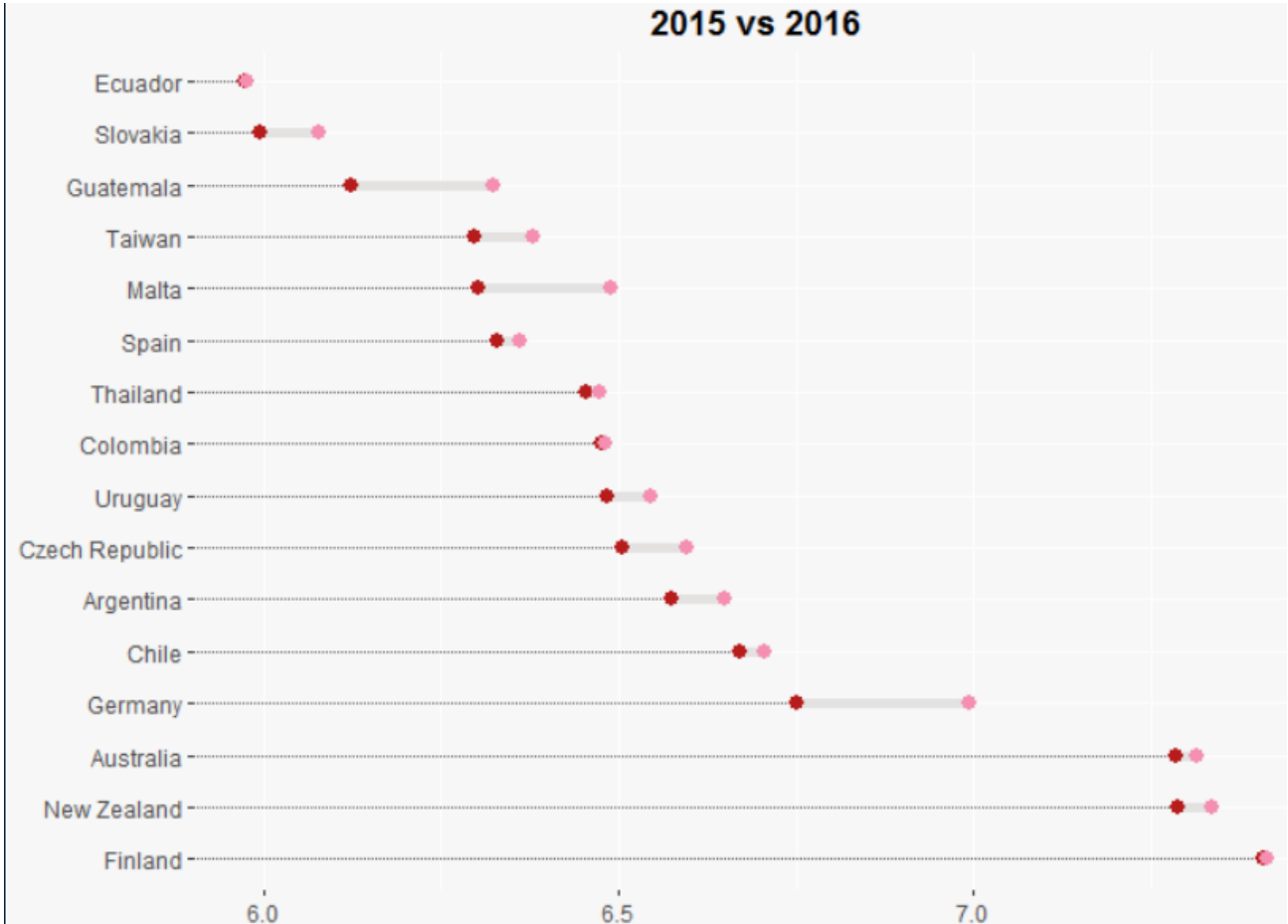
# 03. 나라별 행복점수 시각화 (ggalt)

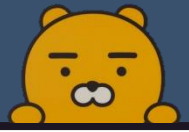



# 03. 나라별 행복점수 시각화 (ggalt)



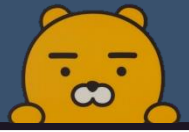

# 03. 나라별 행복점수 시각화 (ggalt)





# 04. 나라별 행복점수 회귀 분석

```
happy2015 <- read.csv("C:₩₩data₩₩happy2015_2.csv",header = T)

normalize<-function(x) {
  return((x-min(x))/ (max(x)-min(x)))
}

happy2015_n <- as.data.frame(lapply(happy2015[,c(4:11)],
                      normalize))

x<-lm(Happiness_Score~Economy_GDP + Family +
      Health + Freedom + Trust + Generosity +
      Dystopia_Residual, data=happy2015_n)
summary(x)
```

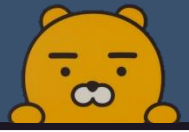


# 04. 나라별 행복점수 회귀 분석

## 2015

```
Coefficients:
      (Intercept)       Economy_GDP            Family            Health
          -0.5287            0.3561            0.2953            0.2159
          Freedom             Trust        Generosity Dystopia_Residual
           0.1410            0.1162            0.1676            0.6895
```

```
Residual standard error: 5.941e-05 on 150 degrees of freedom
Multiple R-squared:      1,     Adjusted R-squared:      1
F-statistic: 3.695e+08 on 7 and 150 DF,  p-value: < 2.2e-16
```



# 04. 나라별 행복점수 회귀 분석

## 2016

```
Coefficients:
      (Intercept)       Economy_GDP            Family            Health
          -0.4516            0.3948            0.2561            0.2062
          Freedom             Trust        Generosity  Dystopia_Residual
           0.1317            0.1093            0.1774            0.6535
```

```
Residual standard error: 6.514e-05 on 149 degrees of freedom
Multiple R-squared:      1,     Adjusted R-squared:      1
F-statistic: 3.206e+08 on 7 and 149 DF,  p-value: < 2.2e-16
```

# 05. 나라별 행복점수 회귀 분석 시각화

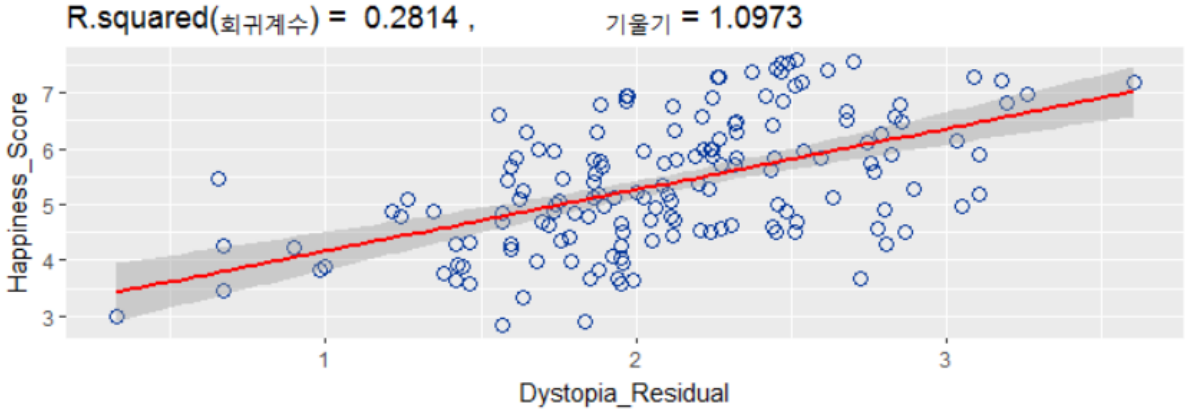
```
library(ggplot2)
library(gridExtra)

ggplotRegression <- function (z) {

ggplot(z$model, aes_string(x = names(z$model)[2], y = names(z$model)[1])) +
        geom_point(shape=1,size=3,color="#003399") +
        stat_smooth(method = "lm", col = "red") +
        labs(title = paste("R.squared(회귀계수) = ",
        signif(summary(z)$r.squared, 5),",                ",
         " 기울기 =", signif(z$coef[[2]], 5)) )
```

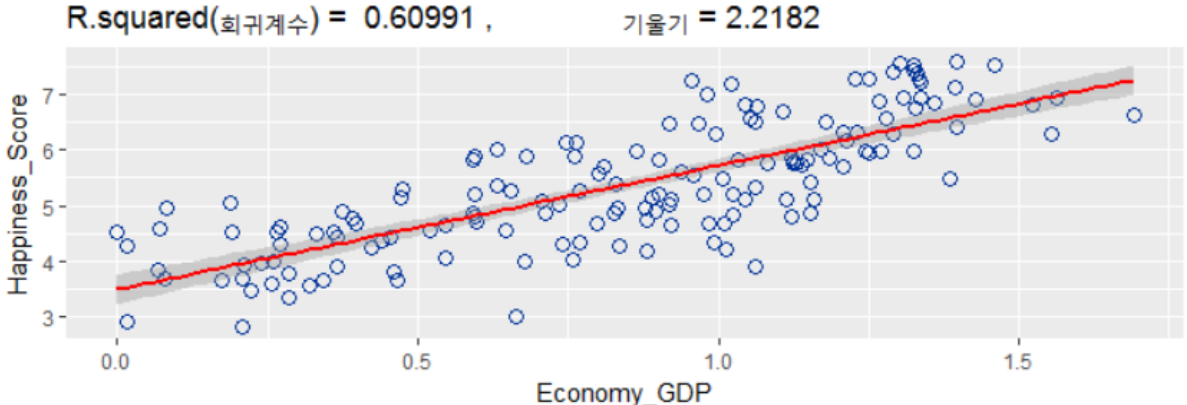
# 05. 나라별 행복점수 회귀 분석 시각화

```
g1<-ggplotRegression(lm(Happiness_Score ~ Dystopia_Residual, data = happy2015))

g2<-ggplotRegression(lm(Happiness_Score ~ Economy_GDP, data = happy2015))

g3<-ggplotRegression(lm(Happiness_Score ~ Family, data = happy2015))

g4<-ggplotRegression(lm(Happiness_Score ~ Health, data = happy2015))

g5<-ggplotRegression(lm(Happiness_Score ~ Generosity, data = happy2015))

g6<-ggplotRegression(lm(Happiness_Score ~ Freedom, data = happy2015))

g7<-ggplotRegression(lm(Happiness_Score ~ Trust, data = happy2015))
```

# 05. 나라별 행복점수 회귀 분석 시각화

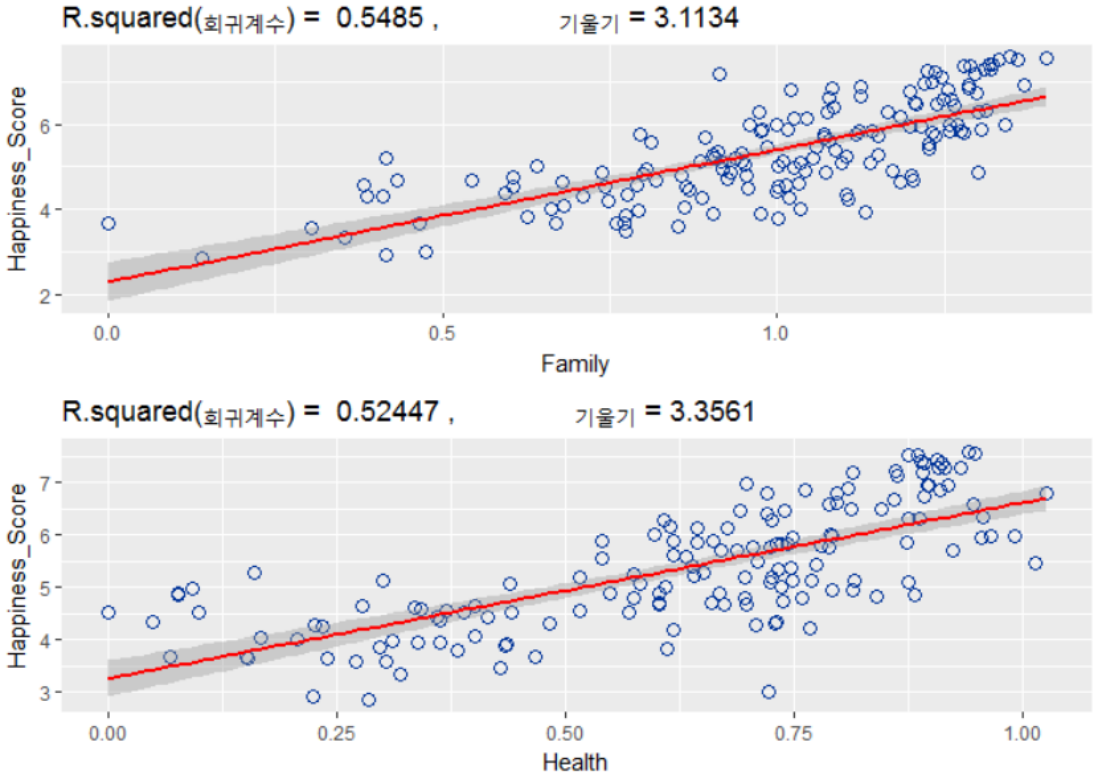
```
grid.arrange(g1,g2, nrow=2)
```



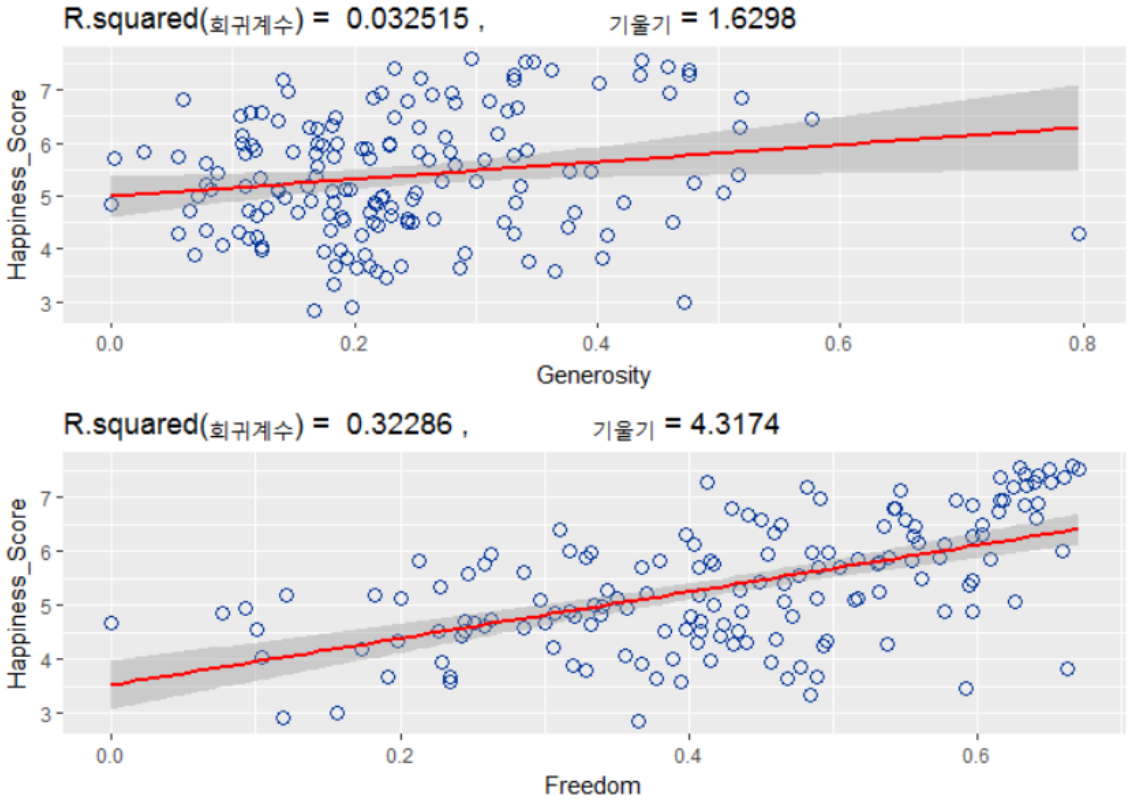
# 05. 나라별 행복점수 회귀 분석 시각화

grid.arrange(g3,g4, nrow=2)



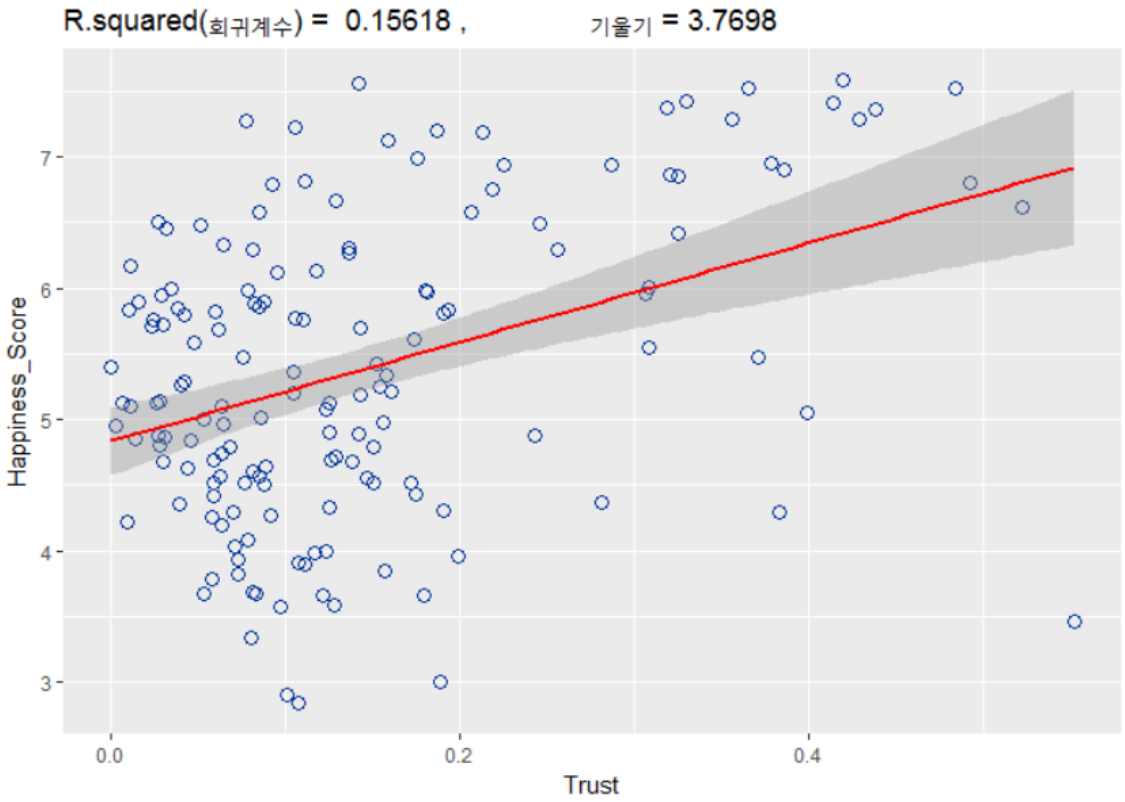
# 05. 나라별 행복점수 회귀 분석 시각화

grid.arrange(g5,g6, nrow=2)



# 05. 나라별 행복점수 회귀 분석 시각화

g7



감사합니다.