# Visualizing Annual Changes in Program Reach Across Several Countries

## Introduction

On my team, one of the (rough) measures we use to assess the impact of a program is how many people that program has reached. This is interesting data that is rarely scrutinized, but recently I was asked to visualize which country programs showed an increase in TR from 2019 to 2020. As the scale of reach varies quite a bit across many countries, I found this to be an interesting data viz challenge that could be standardized and used annually for a quick glance at each country's performance. First let's load in some data, and then we will generate sample data to visualize. We'll just need ggplot and dplyr from the tidyverse package for this one.

## Generate Sample Data

For this exercise, we just need the country variable, 2019 and 2020 data. For data protection purposes, I decided to generate sample data rather than use existing data. You can use `sprintf` to generate a numbered variable, which in this case will create sample countries numbered from 1 to 8. Then, use `sample` to generate random 8 random numbers for 2019 and 8 random numbers for 2020. Pipe all this into a data frame using `data.frame`

```
set.seed(3232021)

country <- c(sprintf("country%01d", seq(1,8)))

tr_2019 <- sample(0:10000, 8, replace=TRUE)

tr_2020 <- sample(0:10000, 8, replace=TRUE)

tr <- data.frame(country, tr_2019, tr_2020)
```

## Transform Data

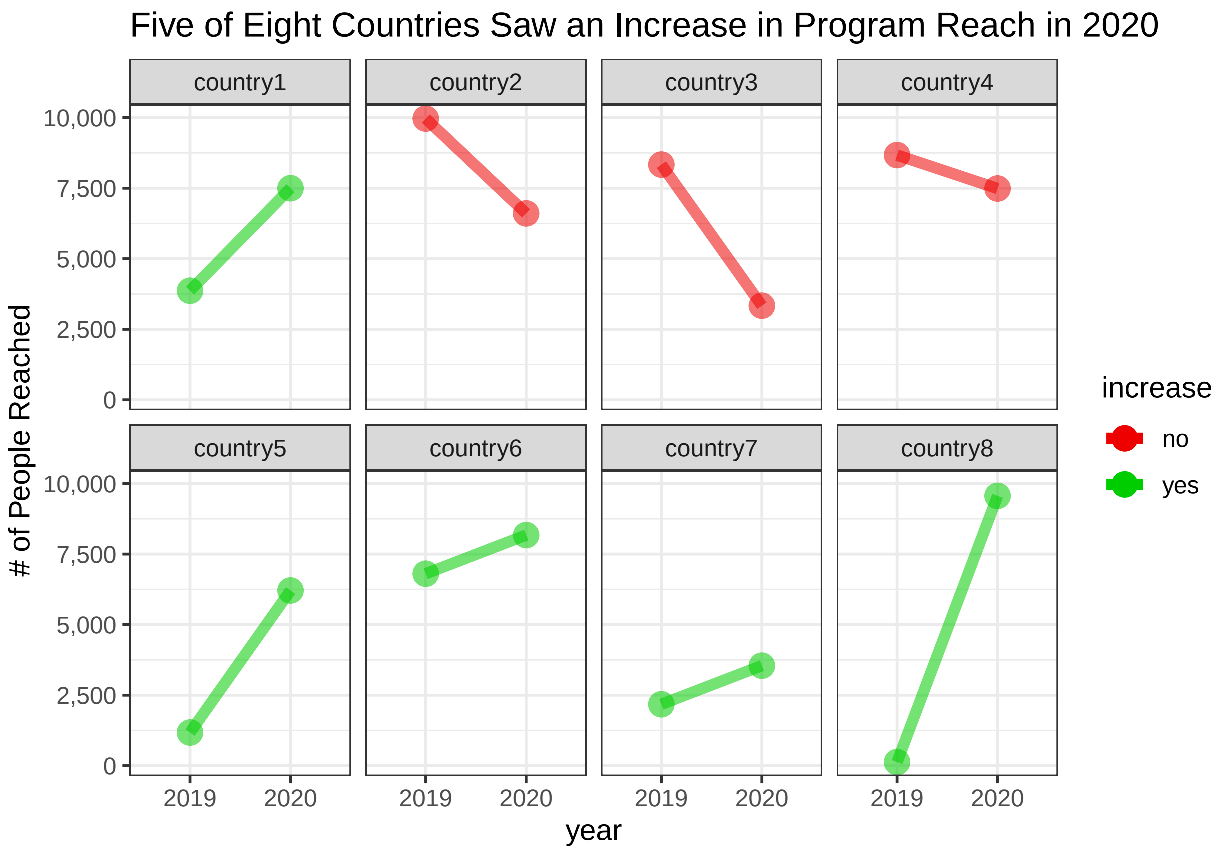
There are two main steps for the data here. The first is the addition of a column that will represent whether or not there was an increase in program reach from 2019 to 2020. You can use `mutate` to create a new column and `ifelse` to essentially say, "If the 2020 column is greater than the 2019 column, generate a 'yes' observation, otherwise generate a 'no' ". The second step is to use `gather` to transform the data for visualization by creating 'year' and 'reach' variables from the columns with the annual reach data. I also renamed my data, just to make it look cleaner on the y axis in the visualization.

```
tr_slope <- tr %>%
  mutate(increase = ifelse(tr$‘tr_2020‘ > tr$‘tr_2019‘, "yes", "no")) %>%
  rename(’2020’ = tr_2020, ’2019’ = tr_2019) %>%
  gather(year, reach, 2:3)
```

## Visualize Data

The final step is to create a faceted slop graph. I originally tried to visualize this on one graph, but because my data had 22 countries with vastly different scales, it came out looking a bit messy. So, I decided to facet the data by country. If you have just a few observations, you may want to consider putting all observations on one graph, though. I think it is also important to create bold colors to represent a “yes” or “no” increase. I chose red and green base colors, which tend to pop out well. You can play around a bit to make the graph look a bit nicer, though I recommend keeping the gridlines so that your audience can garner a quick estimate of the change.

```
ggplot(tr_slope, aes(x = year, y = reach, group = country)) +
  geom_line(aes(color = increase, alpha = 1),  size = 2) +
  geom_point(aes(color = increase, alpha = 1),  size = 4) +
  scale_alpha(guide = ’none’) +
facet_wrap(. ~ country, ncol = 4) +
  scale_color_manual(values=c("red2", "green3")) +
theme(axis.text.x = element_text(size = 10, angle = 45, hjust = 1))+
scale_y_continuous(labels = scales::comma) +
ggtitle("Five of Eight Countries Saw an Increase in Program Reach in 2020") +
theme(plot.title = element_text(hjust = 0.5, margin=margin(10,0,15,0))) +
ylab("# of People Reached")+
theme(axis.title.y = element_text(margin = margin(t = 0, r = 20, b = 0, l = 0)))+
theme(axis.title.x = element_text(margin = margin(t = 10, r = 0, b = 10, l = 0))) +
 theme(panel.border     = element_blank()) +
theme_bw()
```


Five of Eight Countries Saw an Increase in Program Reach in 2020
country1
country2
country3
country4
country5
country6
country7
country8
# of People Reached
year
increase
no
yes