SPEC Lab REU R Resources: Data visualization with `ggplot2`: Scatterplots & Aesthetics

Alix Ziff, Gaea Morales, Zachary Johnson, and Jasmine Chu
based on earlier materials by Therese Anders

Summer 2022

ggplot2 (continued) Scatterplots

The Data Visualization walkthrough is divided into 4 parts: Part 1 (this module), will cover scatterplots. Part 2 will cover line plots. Part 3 will cover map plots. Finally, Part 4 will cover bar plots.

This module covers 2-variable plots, sometimes with a third variable in play as a way to distinguish between different groups in the data. This first walk-through covers scatterplots and their aesthetics. Use this link as a reference for additional commands, aesthetic options, and reference.

Note: Please review the pre-requisite material posted in Module 4 Data Visualization 1, which serves as a great foundation for first-time users of `ggplot2`. This current module builds on the content learned from the prior module.

Multivariate graphs

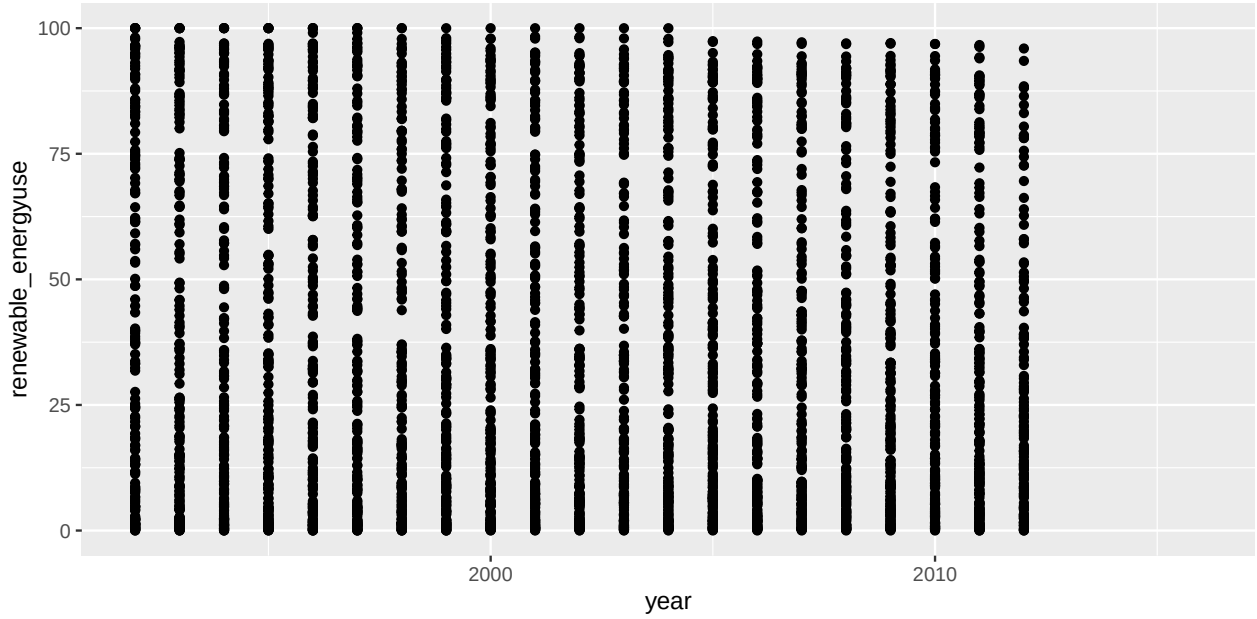
In the previous Data Visualization module, we talked about univariate data summary graphs like kernel density plots, histograms, bar plots, and maps. Similar to working on data collection in excel where our rows are observations and columns are variables, we plotted the range of values of a variable on the x-axis and the density or count of observations on the y-axis.

In this second session, we will be working with multivariate data and will plot two variables–one on the x- and one on the y-axis–against each other with the option of differentiating by the value of additional variable(s), such as through color, shapes, or other aesthetics.

Scatterplots

Scatterplots show the relationship between two variables with the help of points (or other shapes). In `ggplot2` we use the `geom_point()` geometric object to create scatterplots. As an example, we plot the evolution of the worldwide renewable energy usage over time. To begin, once again, make sure that you have set your working directory and loaded the corresponding dataset.

```
dat <- read.csv("wdi_cleaned_part2.csv")

library(ggplot2) # load the necessary libraries
library(dplyr)

ggplot(dat, aes(x = year, y = renewable_energyuse)) +
  geom_point() #we plot our dataset dat with the x axis set to years and the y axis
```




```
# set to level of renewable energy use.
table(dat$year) #this will list the years and the count of how many observations we have
```

```
##
## 1992 1993 1994 1995 1996 1997 1998 1999 2000 2001 2002 2003 2004 2005 2006 2007
##  217  217  217  217  217  217  217  217  217  217  217  217  217  217  217  217
## 2008 2009 2010 2011 2012 2013 2014 2015 2016
##  217  217  217  217  217  217  217  217  217
```

```
# per year.
```

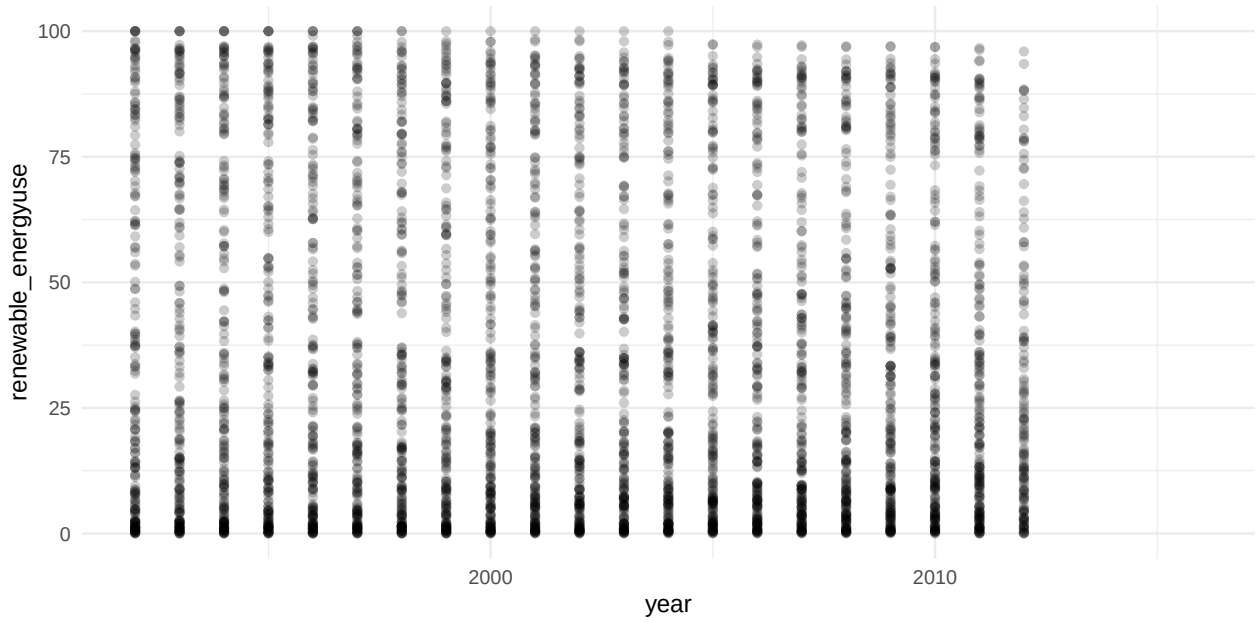
There are 217 observations per year. Due to overplotting, it is hard to draw conclusions from the plot. We have a number of ways in which we can adjust the appearance of the graph to highlight global trends in the data.

Adjusting the opacity of points

One way to use overplotting actively to highlight trends in the data is to reduce the opacity of points. Points will still be plotted on top of each other, but overlaying multiple transparent points will create clusters that signal an agglomeration of data points.

In addition, we use a theme with a white background to further increase the visibility of clusters in the data. In this example, while reducing the opacity of points does not significantly aid our understanding of over-time trends in the data, the plot shows that in the majority of countries between 0% to 10% of the energy used comes from renewables.

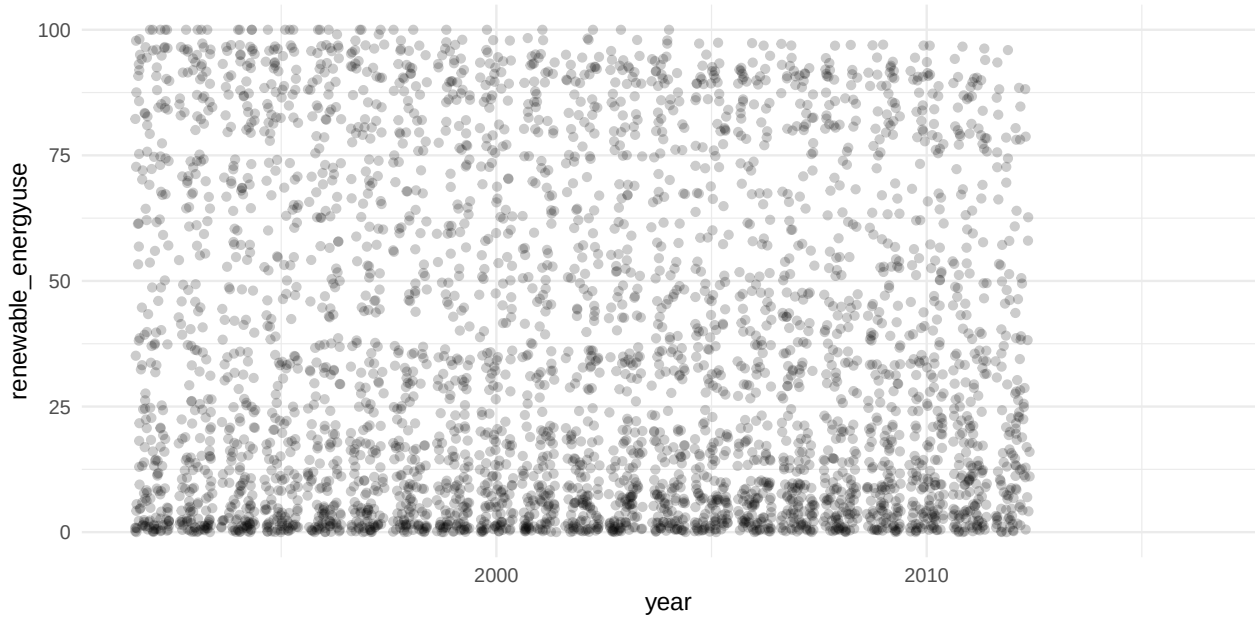
```
ggplot(dat, aes(x = year, y = renewable_energyuse)) +
  geom_point(alpha = 0.2) + #this sets our opacity
  theme_minimal() #this sets our background to white
```



Jitter

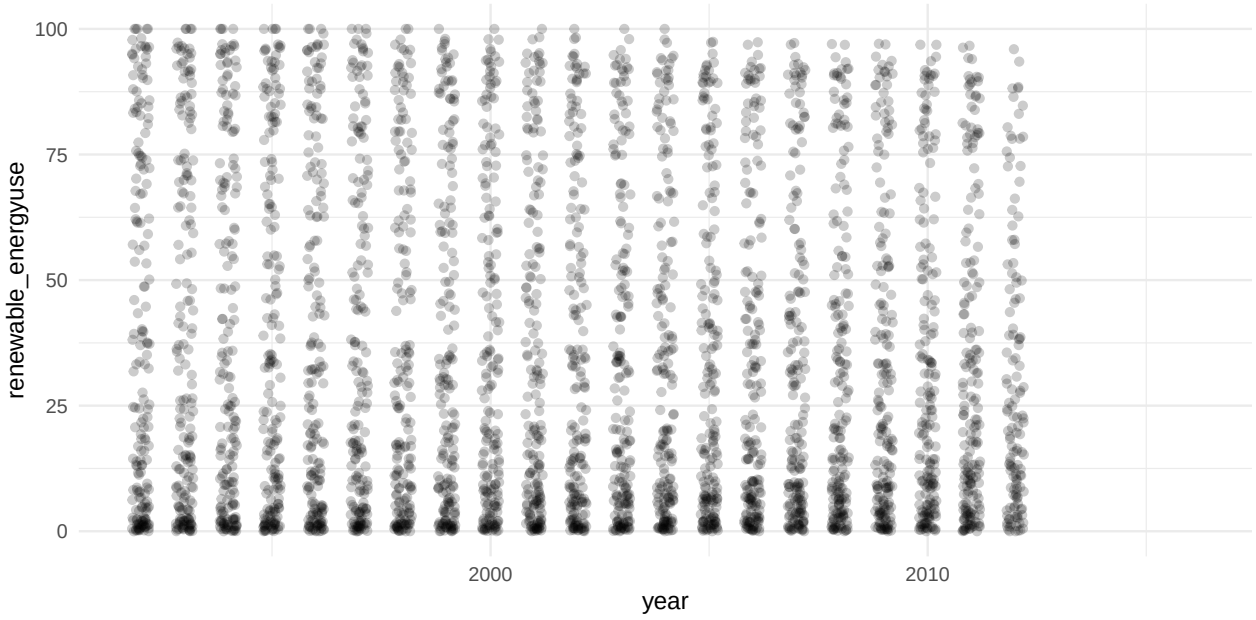
Jittering points is another way to reduce the negative effects of overplotting. The `position_jitter()` argument randomly adds small values to each point. This way, each point is randomly shifted a tiny bit to the top, right, bottom, or left. Note that unless you set a seed (i.e., using the set.seed() function) to control the randomness, a graph with a jitter function will appear a little bit different every time you execute it.

```
ggplot(dat, aes(x = year, y = renewable_energyuse)) +
  geom_point(alpha = 0.2, position = position_jitter()) + #this sets our opacity and
  # jitters points
  theme_minimal() #this sets our background to white
```



We can change the default jitter value with the `width` and `height` parameters inside the `position_jitter()` argument. The default is for the data points to occupy 80% of the implied bins (see http://ggplot2.tidyverse.org/reference/geom_jitter.html). In our case, by decreasing the jitter width, we can increase the visual separation between years.

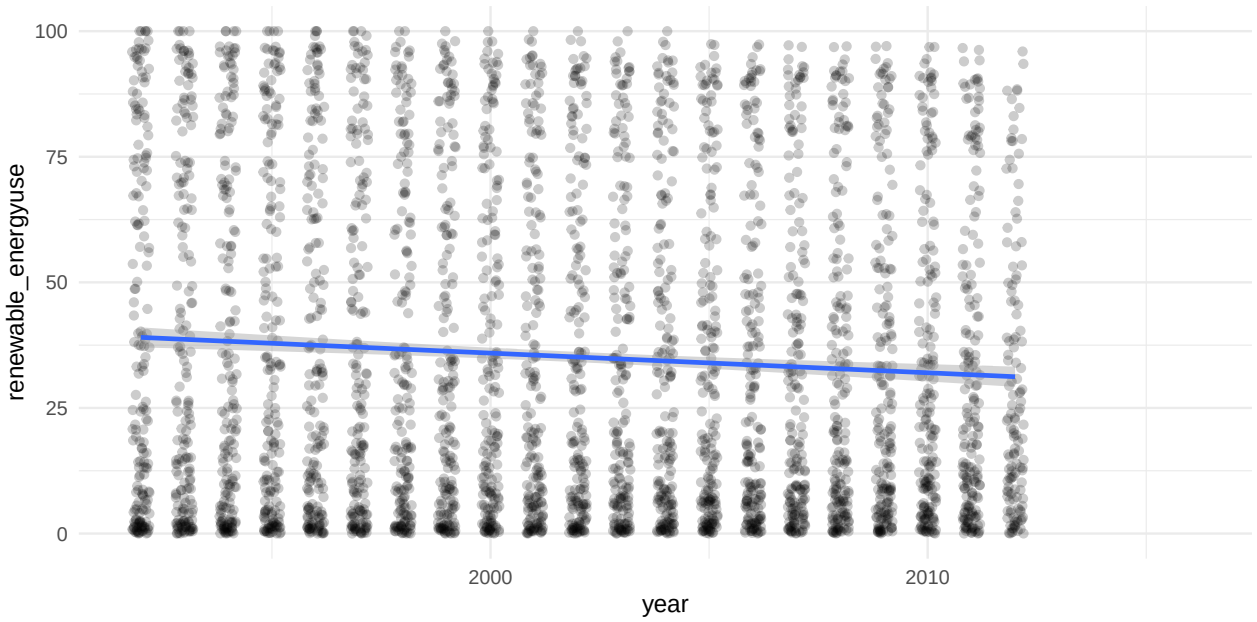
```
ggplot(dat, aes(x = year, y = renewable_energyuse)) +
  geom_point(alpha = 0.2, position = position_jitter(width = 0.2)) + #changed the
  # default jitter value
  theme_minimal()
```



Adding trend lines

Another way to highlight trends in the data is to overlay a scatterplot with a trend line. In `ggplot2` this is implemented using the `stat_smooth()` function. For example, we could overlay the graph with the line of best fit of a linear model that regresses the proportion of renewable energy usage on the year using the `method = "lm"` argument. The resulting graph suggests that globally, the usage of renewables has decreased over time.

```
ggplot(dat, aes(x = year, y = renewable_energyuse)) +
  geom_point(alpha = 0.2, position =  position_jitter(width = 0.2)) + #jittered points
  theme_minimal() + #white & simplified background
  stat_smooth(method = "lm") #creating our trend line
```



Note: `method =` in `stat_smooth()` specifies the parameters of the smooth statistic. Notice that we specify the method as "lm", because we are dealing with a linear model. If using a generalized linear model, specify "glm". You can see other options for smoothing parameters here.

Using color to distinguish groups

Suppose we wanted to know whether the evolution in the usage of renewable energies differs between richer and poorer countries. We could do this by creating a binary variable (or dummy) that codes the wealth of countries and passing this variable to the color aesthetic. Here, we will use the median per capita GDP value as a cut point to split the sample into two roughly equally sized groups. Another alternative would be to use the average per capita GDP as a cut-off.

```
str(dat$gdppc) #looking at our GDP variable
```

```
##  num [1:5425] NA 3014 9817 NA NA ...
```

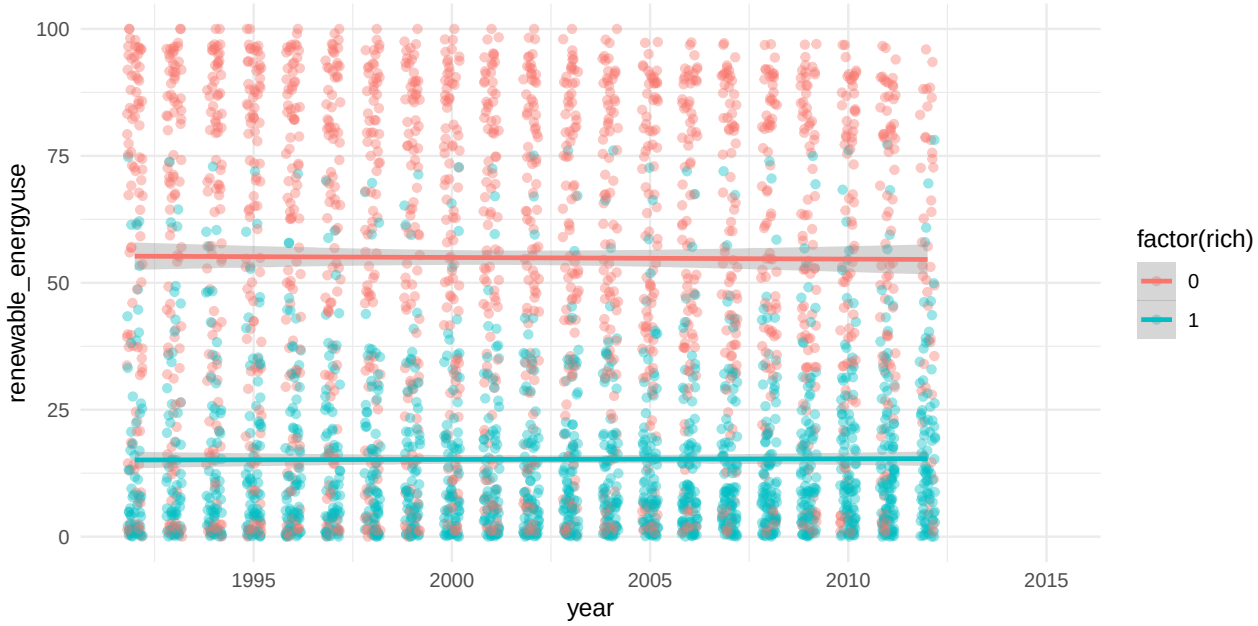
```
dat$rich <- ifelse(dat$gdppc >= median(dat$gdppc, na.rm = T), 1, 0) # re-writing our
  # dat$rich object with our GDP values that are greater than or equal to the
  # median of our GDP per capita variable values

table(dat$rich)
```

```
##
##    0    1
## 2233 2234
```

By passing our new `rich` variable to the color aesthetic, all following geometric objects will be plotted for both groups. This means that both the `geom_point()` objects and the `stat_smooth()` objects will be plotted for rich and poor countries. There are a number of missing values in our new variable `rich`. To avoid plotting these as a separate group, we can subset the data to not include missing values on the variable `rich` using `subset()` before the aes() argument.

The plot suggests that rich countries have lower levels of renewable energy usage. Interestingly, on average, the percentage of energy that comes from renewables has not changed much over time for either the rich or the poor countries.

```
ggplot(subset(dat, !is.na(rich)), # excluding NAs from our rich variable while plotting
       # the data (not removing them from the dataset itself)
       aes(x = year,
           y = renewable_energyuse,
           color = factor(rich))) + #plotting the x-axis with years and y-axis with renewable energy us
  geom_point(alpha = 0.4, position = position_jitter(width = 0.2)) +
  theme_minimal() +
  stat_smooth(method = "lm")
```

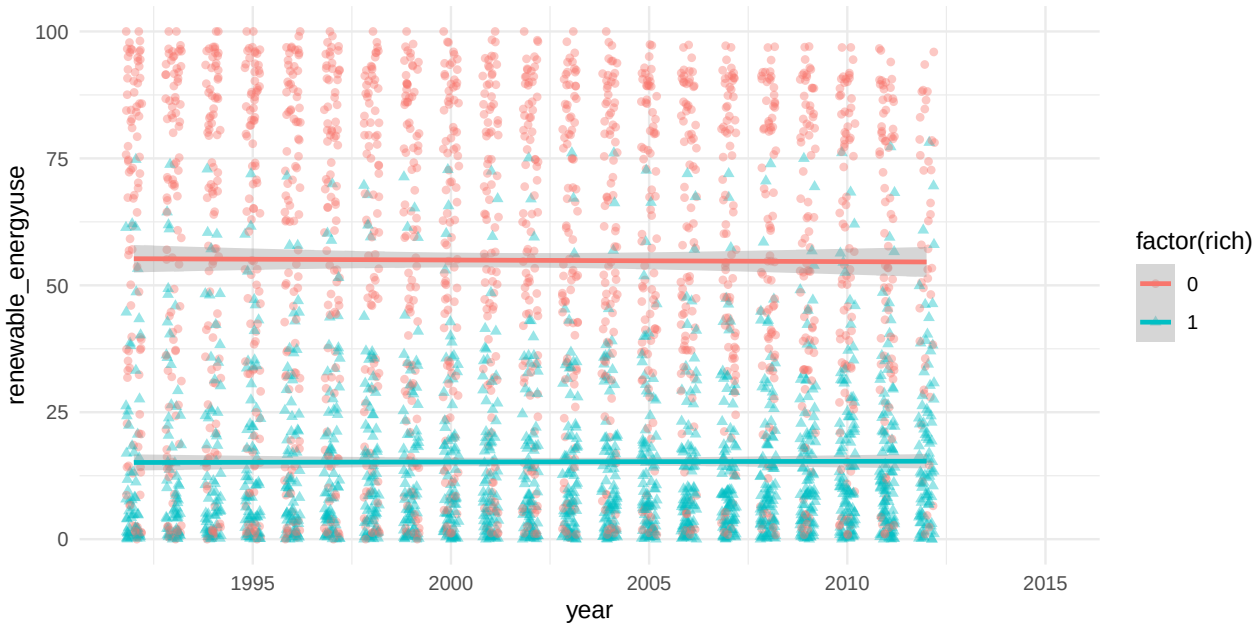


Helpful hint: Note that we wrap the `rich` variable in the `factor()` function so that ggplot knows that our dummy variable is a factor variable, and not a continuous variable. When plotting color with a continuous variable, we would end up with a color scale (e.g., dark blue –> light blue) as opposed to different, more distinguishable colors (e.g., red and blue).

Using shapes to distinguish groups

Sometimes, using color is not the optimal choice to distinguish groups, for example if we have to create plots on a grey scale or want to make sure our graphs are color blind safe. We can instead use shapes to distinguish groups, or combine the distinction of shapes and color like in the example below.

```
ggplot(subset(dat, !is.na(rich)), #if the value is not (! = not) NA in our subset
       # then it will be plotted with the following aesthetics:
       aes(x = year,
           y = renewable_energyuse,
           color = factor(rich), #changing the color
           shape = factor(rich))) + #changing the shape
  geom_point(alpha = 0.4, position = position_jitter(width = 0.2)) + #opacity and jittering
  theme_minimal() + #white background
  stat_smooth(method = "lm")
```

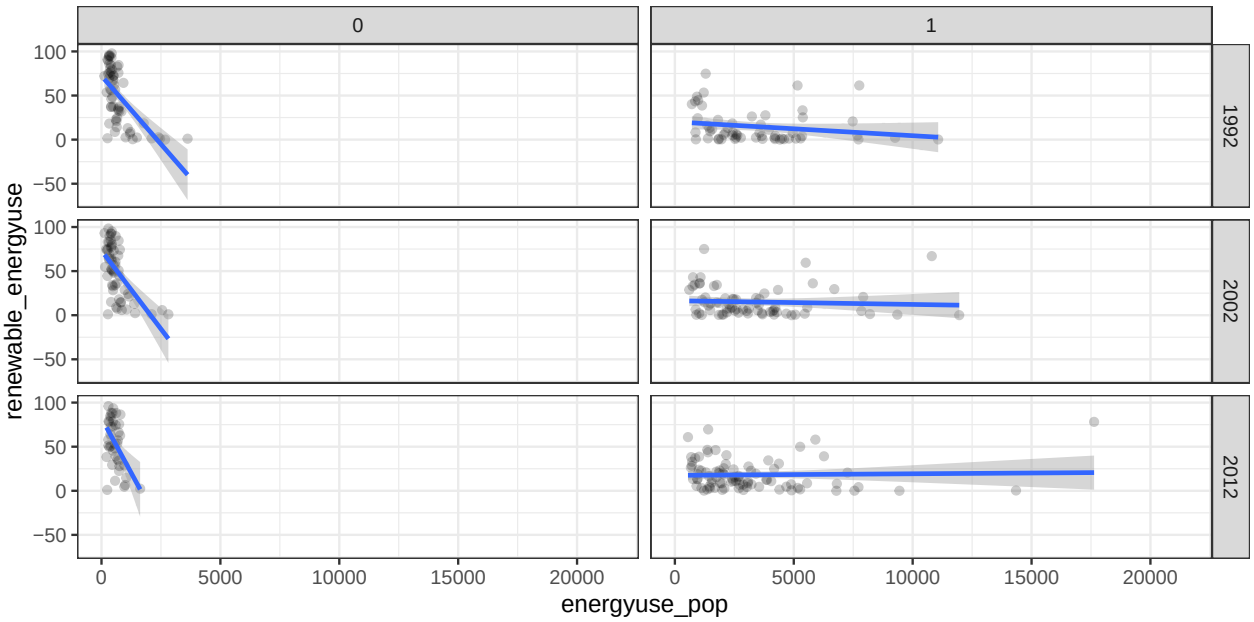


Using faceting to distinguish groups

We can also plot separate scatterplots for each group using faceting. In the first part of this workshop, we used the `facet_wrap()` function which allows us to plot relationships across groups contained in a single variable. Today, we will instead use the `facet_grid()` function which allows us to plot the relationship across two variables.

Suppose we want to know whether there is a difference in the relationship between per capita energy consumption and percentage of renewables in poorer versus richer countries, **and** how the relationship changes over time. We will only plot the relationship for years in which data are available on the per capita energy consumption: 2000, 2010, and 2012.

```
ggplot(subset(dat, !is.na(rich) & year %in% c(1992, 2002, 2012)), #plotting the years 1992,
       # 2002, and 2012
       aes(x = energyuse_pop,
           y = renewable_energyuse)) +
  geom_point(alpha = 0.2) +
  theme_bw() +
  stat_smooth(method = "lm") +
  facet_grid(factor(year)~factor(rich))
```




```
cor.test(dat$energyuse_pop, dat$gdppc) # simple test for association between paired samples
```

```
##
##  Pearson's product-moment correlation
##
## data:  dat$energyuse_pop and dat$gdppc
## t = 82.541, df = 3069, p-value < 2.2e-16
## alternative hypothesis: true correlation is not equal to 0
## 95 percent confidence interval:
##  0.8190053 0.8409938
## sample estimates:
##       cor
## 0.8303224
```

We can draw a number of conclusions from the plot. First, the relationship between per capita energy consumption and the degree to which renewables are used to produce energy does not change much over time. What does influence the relationship between the two variables of interest is the wealth of a country. In richer countries, the amount of energy used and the prevalence of renewables are not strongly related. In poorer countries, the more energy is consumed on average, the less renewables contribute to the production of energy. Note, however, that the wealth of a country and the amount of energy used per inhabitant are positively correlated with a correlation coefficient of 0.83 (pooled across all available observations). This means that on average, the more energy a country consumes per capita, the more similar it will be to a richer country.

We will learn in the following walkthroughs other means of visualizing data.