Working with categorical variables as factors

Author: Nicholas G Reich

Different kinds of variables

Give some examples of each

- Continuous: variables taking any real number value in a range
- Discrete: variables taking an integer value
- Categorical: variables taking one of a fixed set of values

Categorical variables in R often start as strings

By default, characters are read in as characters, not as factors, although you can force factors. A factor is a special type of R data type that can be used to represent a categorical variable with a fixed number of responses.

```
library(tidyverse)
co2 <- read_csv("../../data/co2emissions.csv")
head(co2)
```

```
## # A tibble: 6 x 3
##    Year   CO2 Type
##   <dbl> <dbl> <chr>
## 1  1980  81.2 Rural Diesel
## 2  1981  89.9 Rural Diesel
## 3  1982  89.9 Rural Diesel
## 4  1983  95.7 Rural Diesel
## 5  1984  95.7 Rural Diesel
## 6  1985  95.7 Rural Diesel
```

Tidy aggregation and summary by category

We can use `group_by()` and `summarize()` to aggregate and compute summaries by categories. (You will be asked to do this in a future coding challenge.)
For example, here we compute the average CO2 emissions across all years, for each type of vehicle.

```
co2 %>%
  group_by(Type) %>%
  summarize(mean_emissions = mean(CO2))
```

```
## # A tibble: 4 x 2
##   Type           mean_emissions
##   <chr>                   <dbl>
## 1 Rural Diesel             146.
## 2 Rural Gasoline           390.
## 3 Urban Diesel             127.
## 4 Urban Gasoline           669.
```

Tidy aggregation and summary by category

You can compute multiple summaries at once.

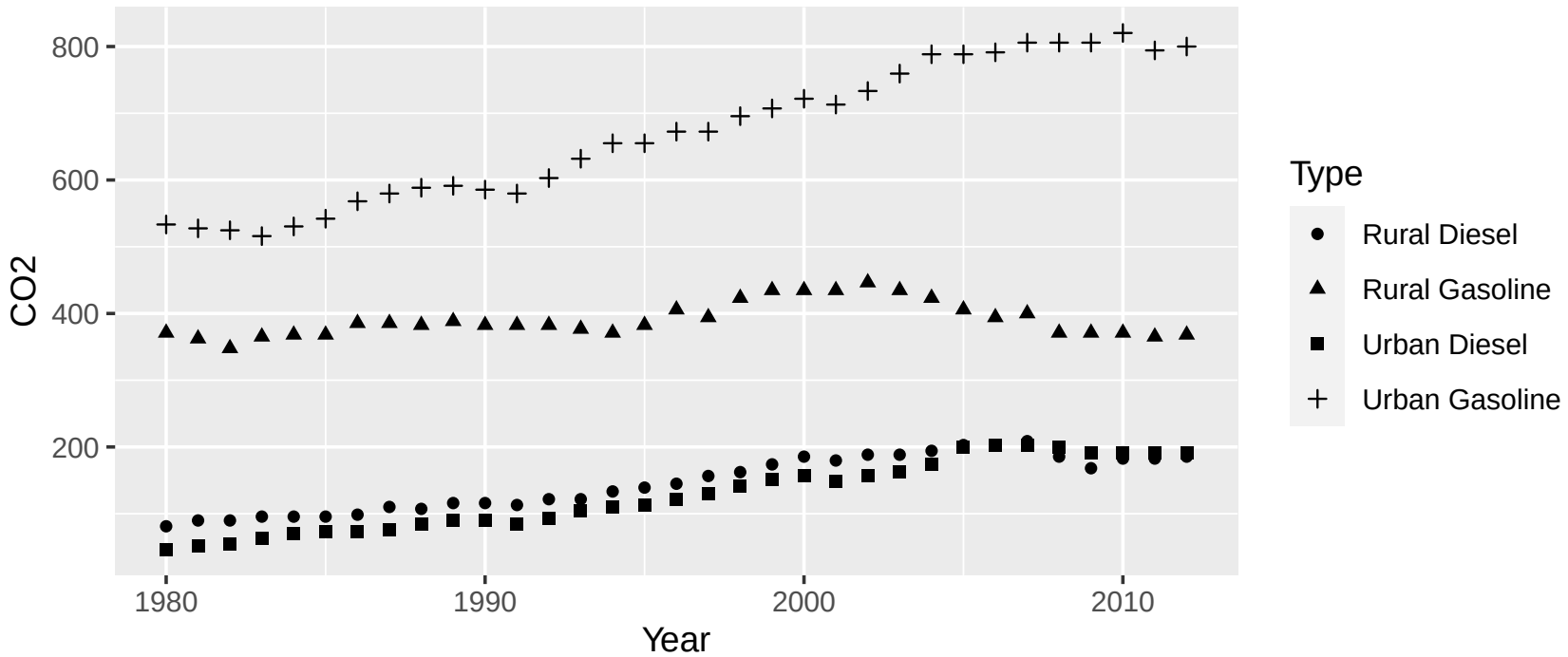
```
co2 %>%
  group_by(Type) %>%
  summarize(
    mean_emissions = mean(CO2),
    max_emissions = max(CO2),
    min_emissions = min(CO2))
```

```
## # A tibble: 4 x 4
##   Type           mean_emissions max_emissions min_emissions
##   <chr>                   <dbl>         <dbl>         <dbl>
## 1 Rural Diesel             146.          209.          81.2
## 2 Rural Gasoline           390.          446.         348.
## 3 Urban Diesel             127.          203.          46.4
## 4 Urban Gasoline           669.          820.         516.
```

Using categorical variables for aesthetics

Note that R translates the character variable into a factor for you without you doing anything.

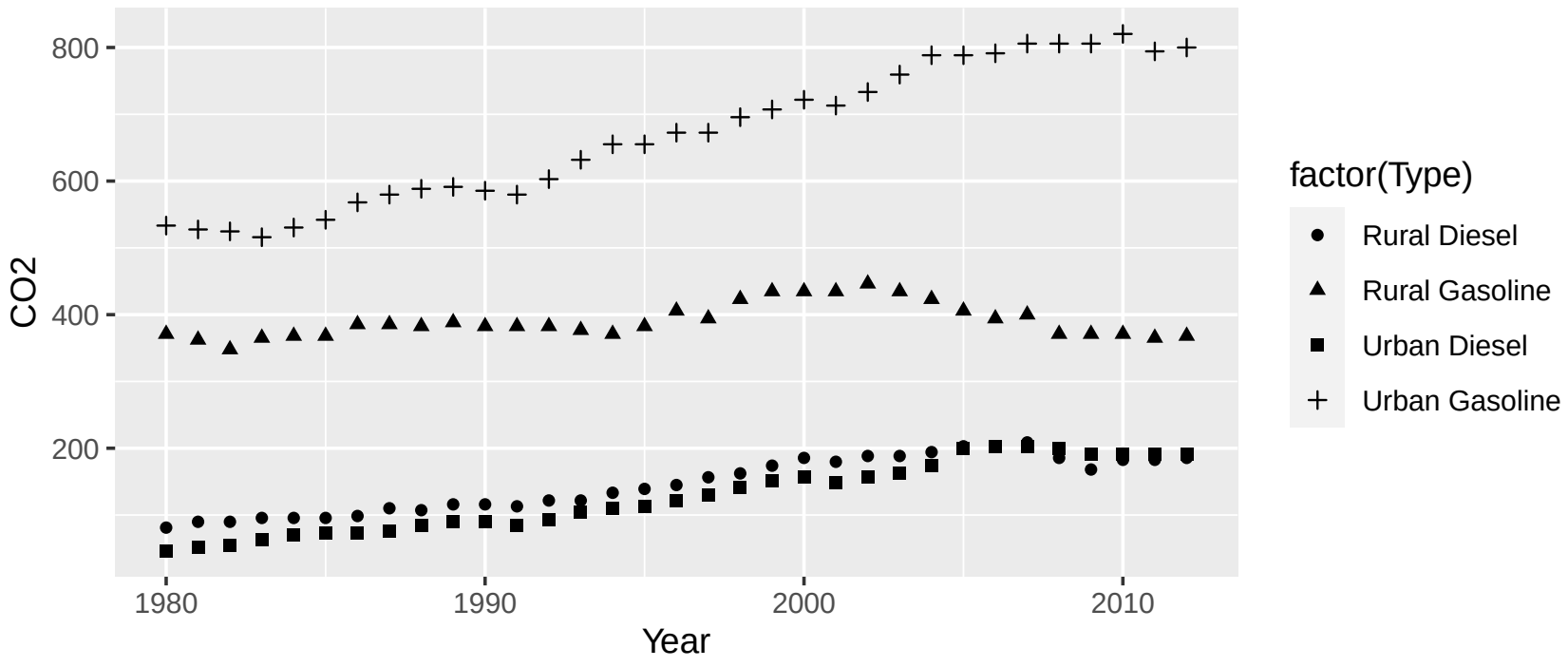
```
ggplot(co2, aes(x = Year, y = CO2, shape = Type, fill = Type))+
  geom_point()
```



Using factors for aesthetics

Note that you can get the same result by explicitly calling Type a `factor`.

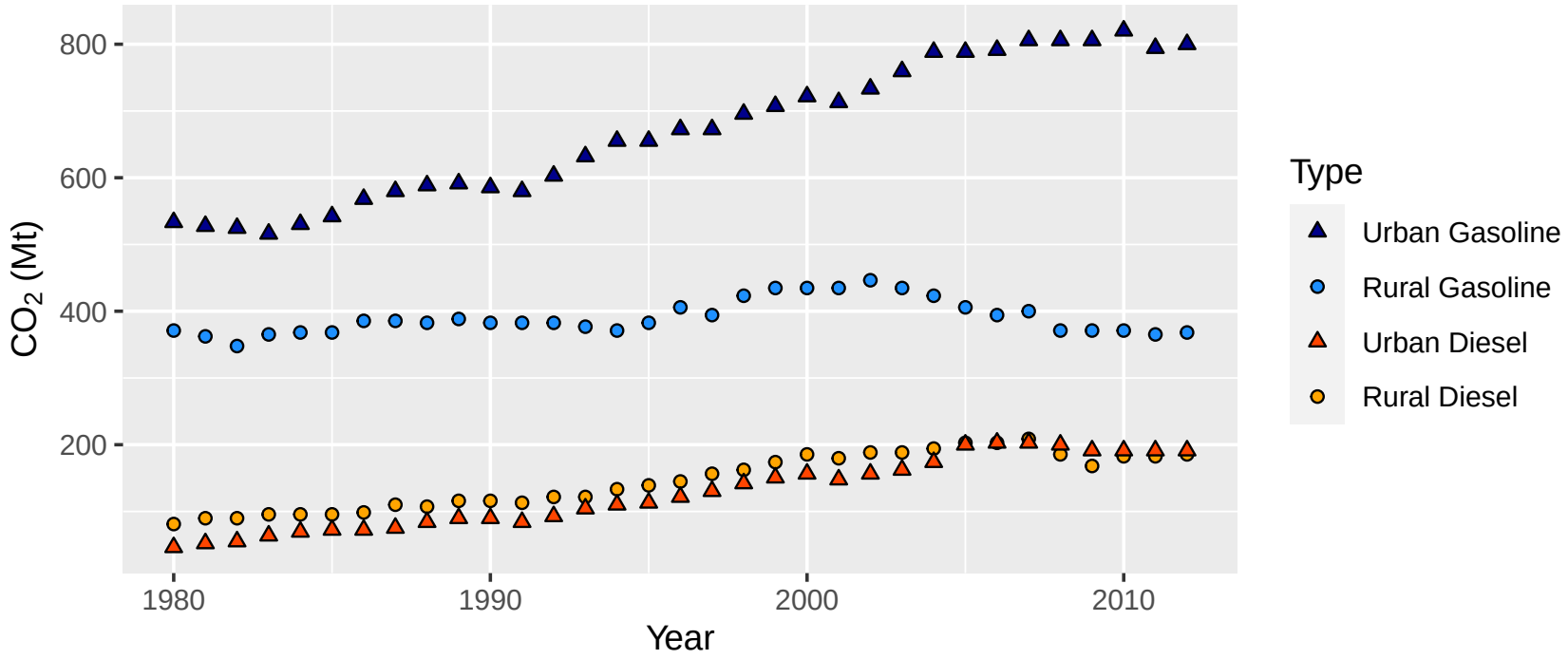
```
ggplot(co2, aes(x = Year, y = CO2, shape = factor(Type), fill = factor(Type)))+
  geom_point()
```



Using factors for aesthetics

And with just a few small tweaks, we can customize

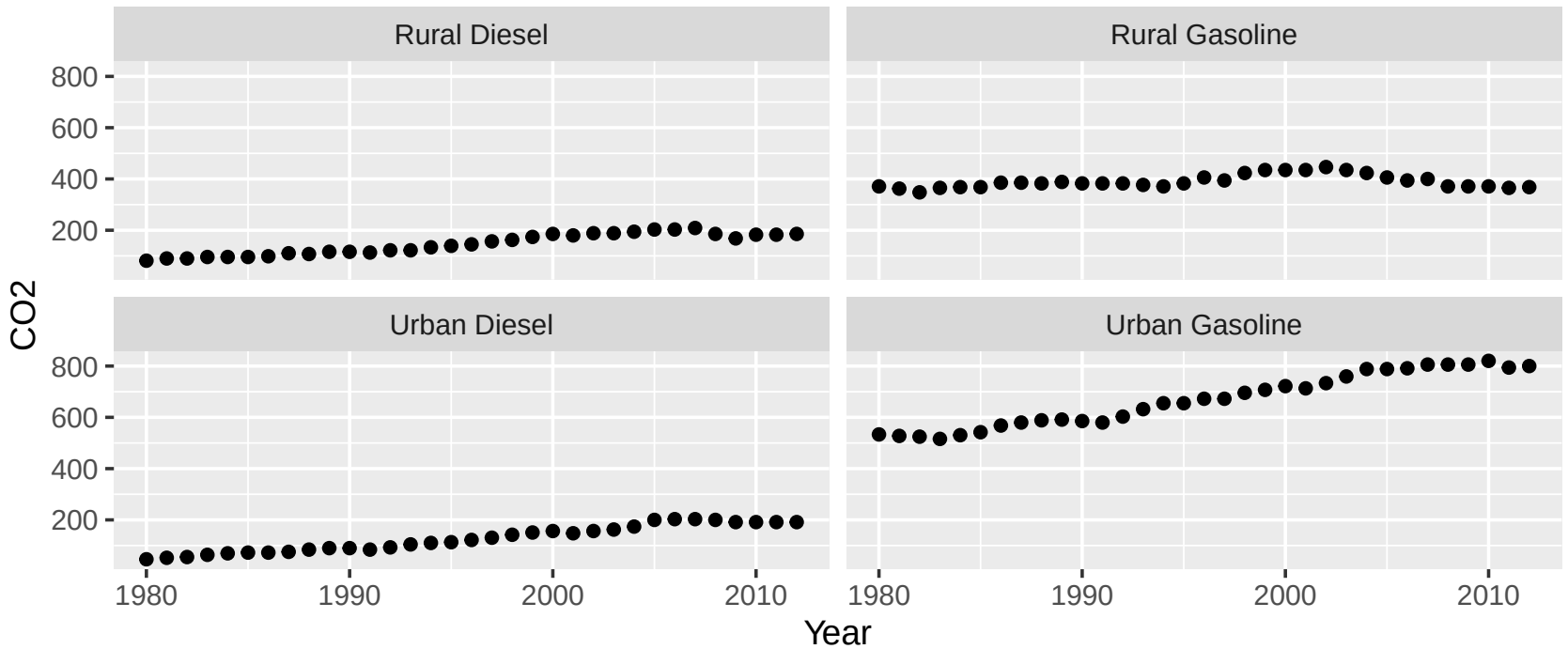
```
levels <- c("Urban Gasoline", "Rural Gasoline", "Urban Diesel", "Rural Diesel")
ggplot(co2, aes(x = Year, y = CO2, shape = Type, fill = Type)) +
  geom_point() +
  scale_shape_manual(breaks=levels, values=c(24, 21, 24, 21)) +
  scale_fill_manual(breaks = levels,
                    values=c("blue4", "dodgerblue", "orangered", "orange")) +
  ylab(expression(CO[2]*" (Mt)"))
```



Using factors for faceting

Factors (or any variable with a small number of distinct values) can be used to create facets as well.

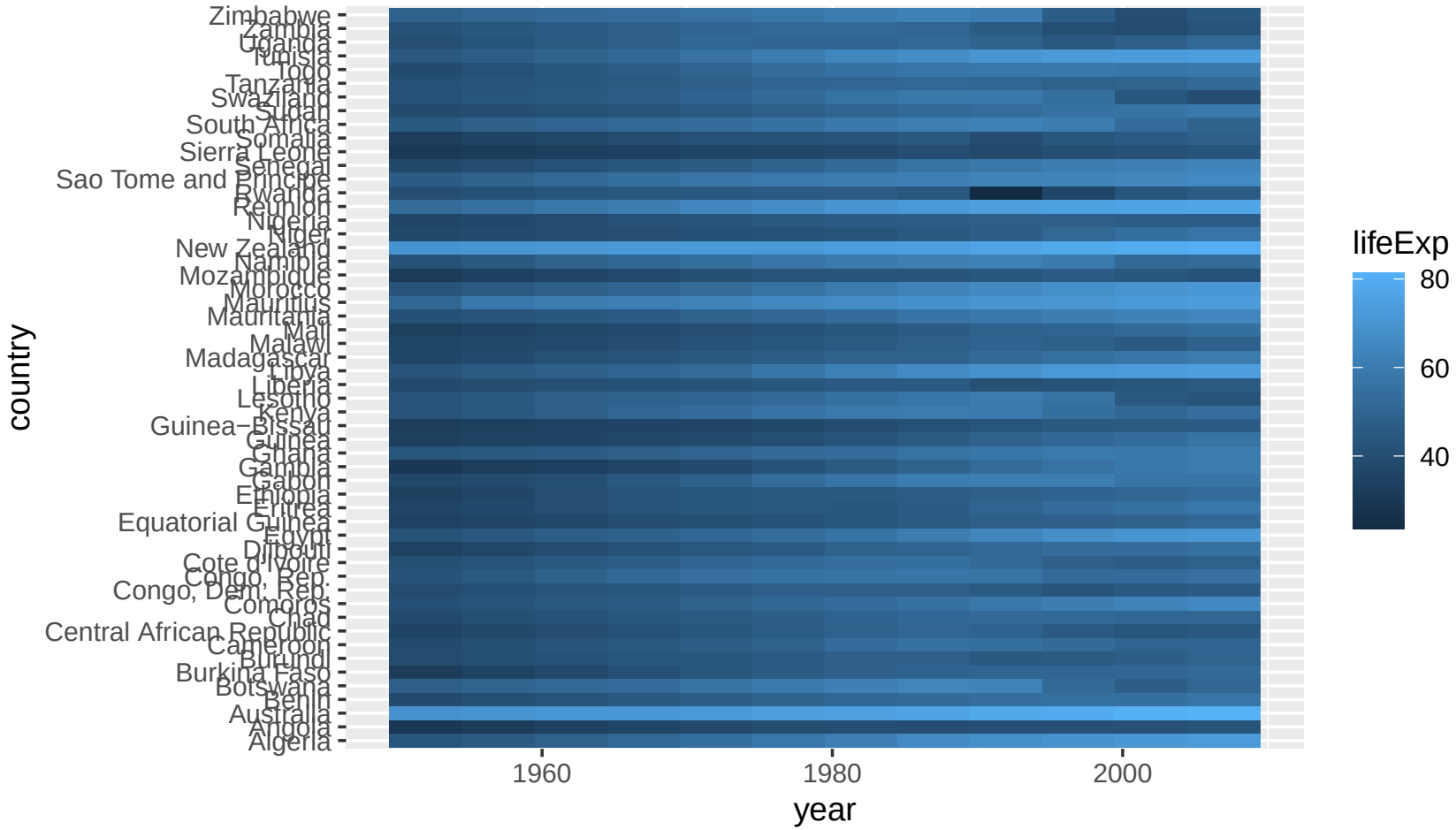
```
ggplot(co2, aes(x = Year, y = CO2)) +
  geom_point() +
  facet_wrap(~Type)
```



Advanced use of factors: ordering

Turning categorical variables into ordered factors might help you show more data.

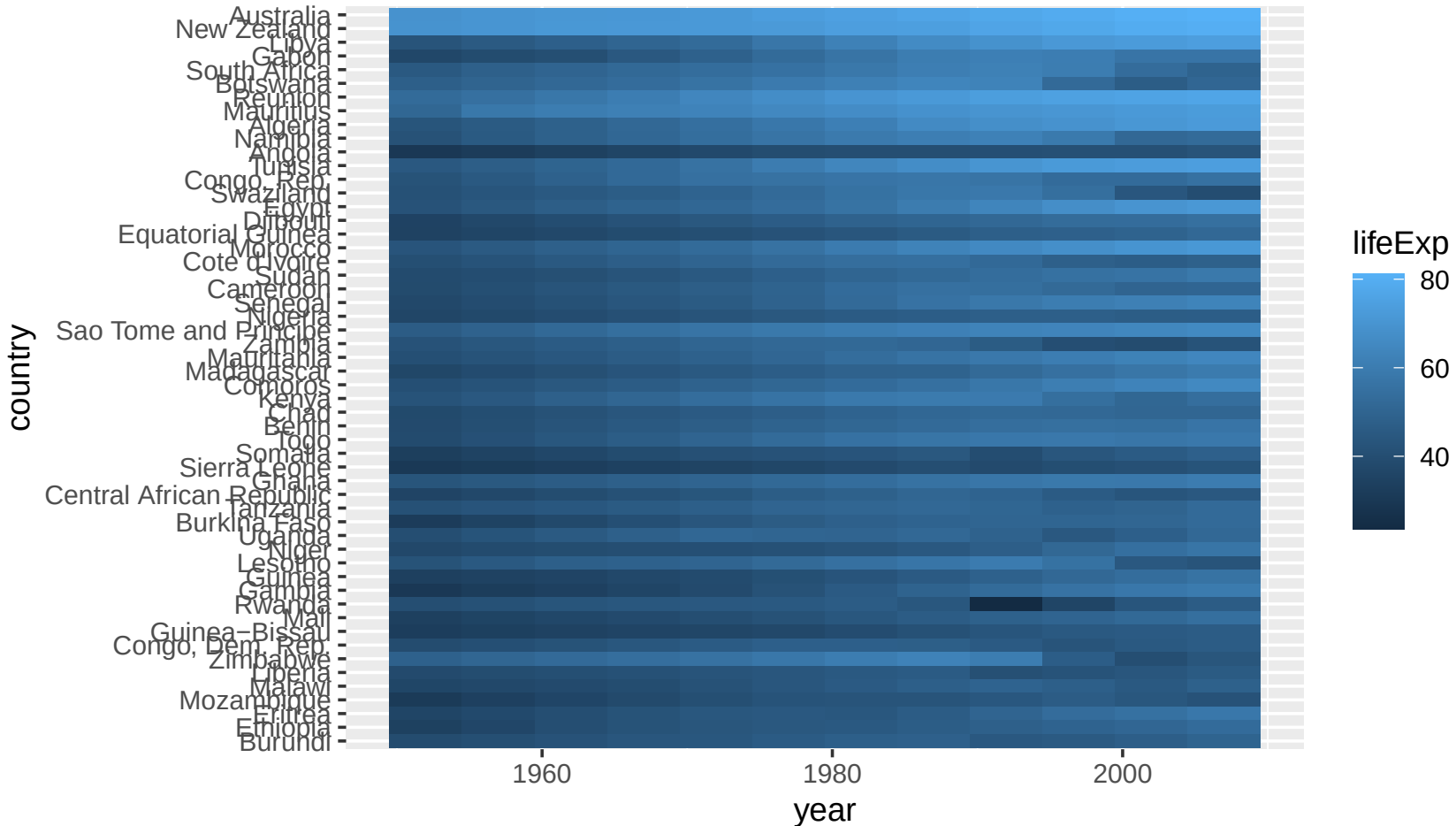
```
gapminder <- read_csv("../../data/gapminder.csv") %>%
    filter(continent %in% c("Africa", "Oceania"))
ggplot(gapminder, aes(x=year, y=country, fill=lifeExp)) +
    geom_tile()
```



Advanced use of factors: ordering

If "order matters" for your categorical variable, then turning it into an ordered factor might be useful.

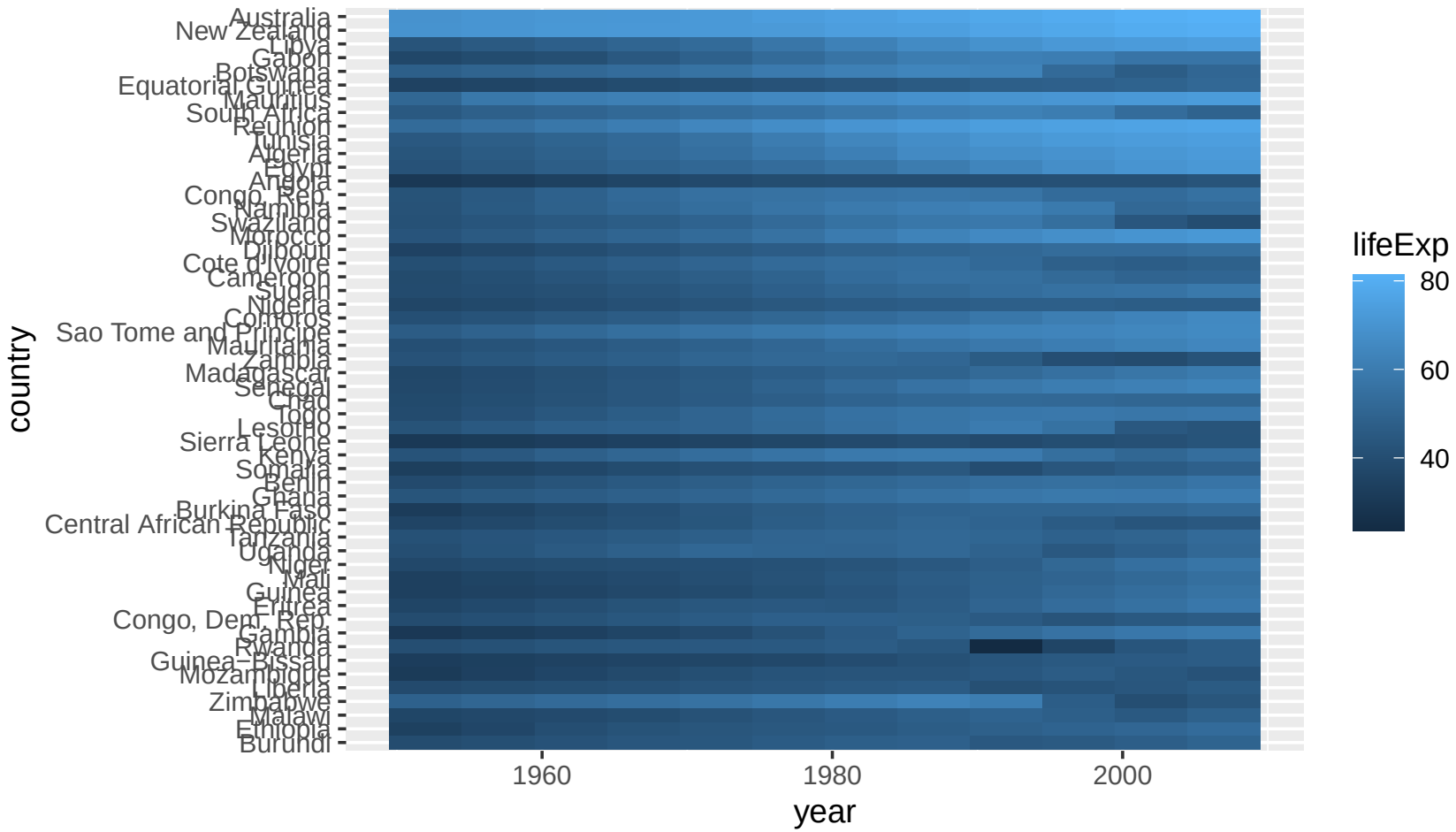
```
## this redefines country based on average GDP
gapminder <- mutate(gapminder, country = reorder(country, gdpPercap, FUN=mean))

ggplot(gapminder, aes(x=year, y=country, fill=lifeExp)) +
    geom_tile()
```



Advanced use of factors: ordering

Here we order based on the maximum GDP rather than the mean.

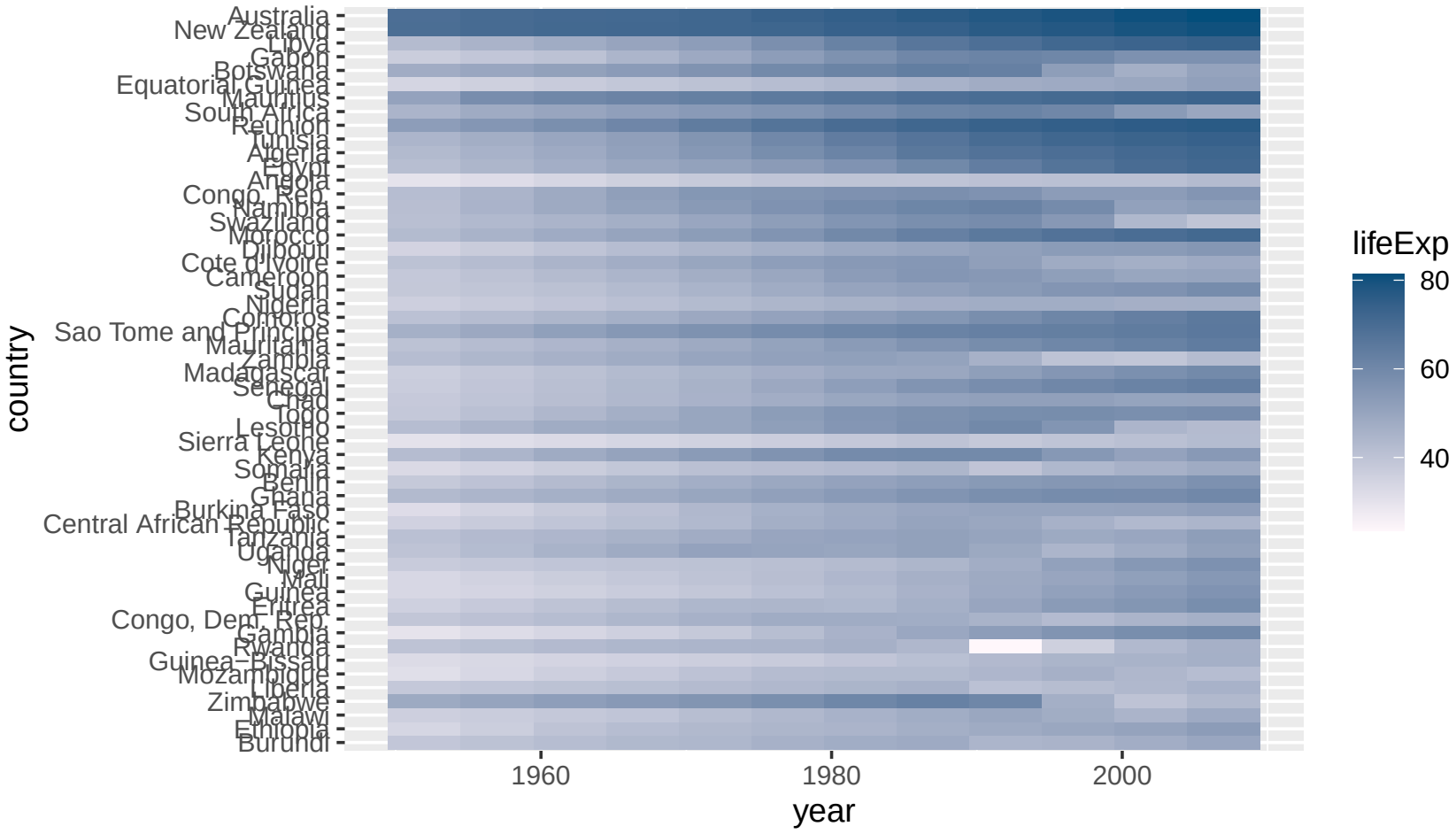
```
## this redefines country based on max GDP
gapminder <- mutate(gapminder, country = reorder(country, gdpPercap, FUN=max))

ggplot(gapminder, aes(x=year, y=country, fill=lifeExp)) +
    geom_tile()
```



Trying out different color scales

Using color scales from ColorBrewer: colorbrewer2.org.

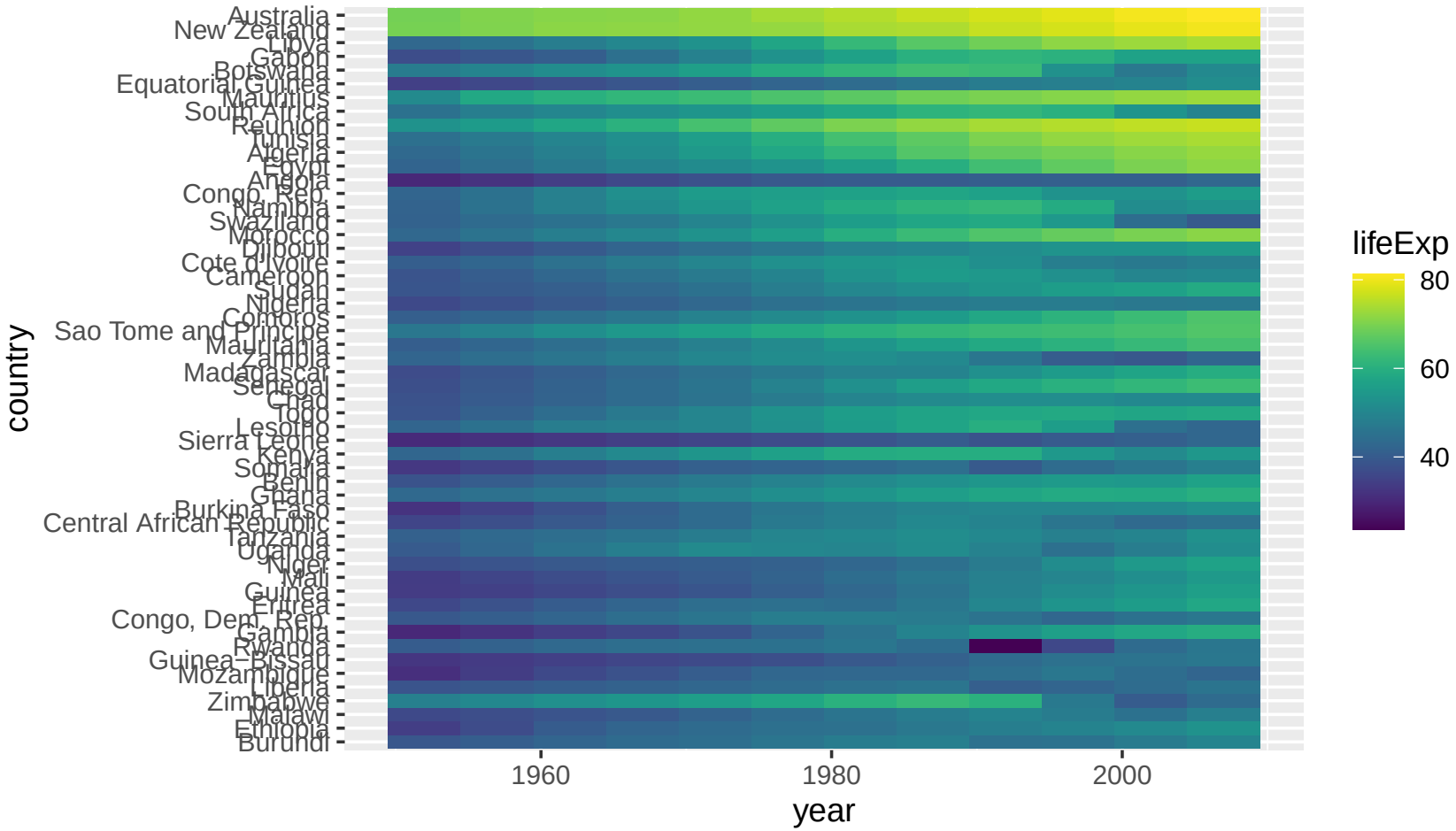
```
ggplot(gapminder, aes(x=year, y=country, fill=lifeExp)) +
    geom_tile() +
    scale_fill_gradient(low="#fff7fb", high="#034e7b")
```



Trying out different color scales

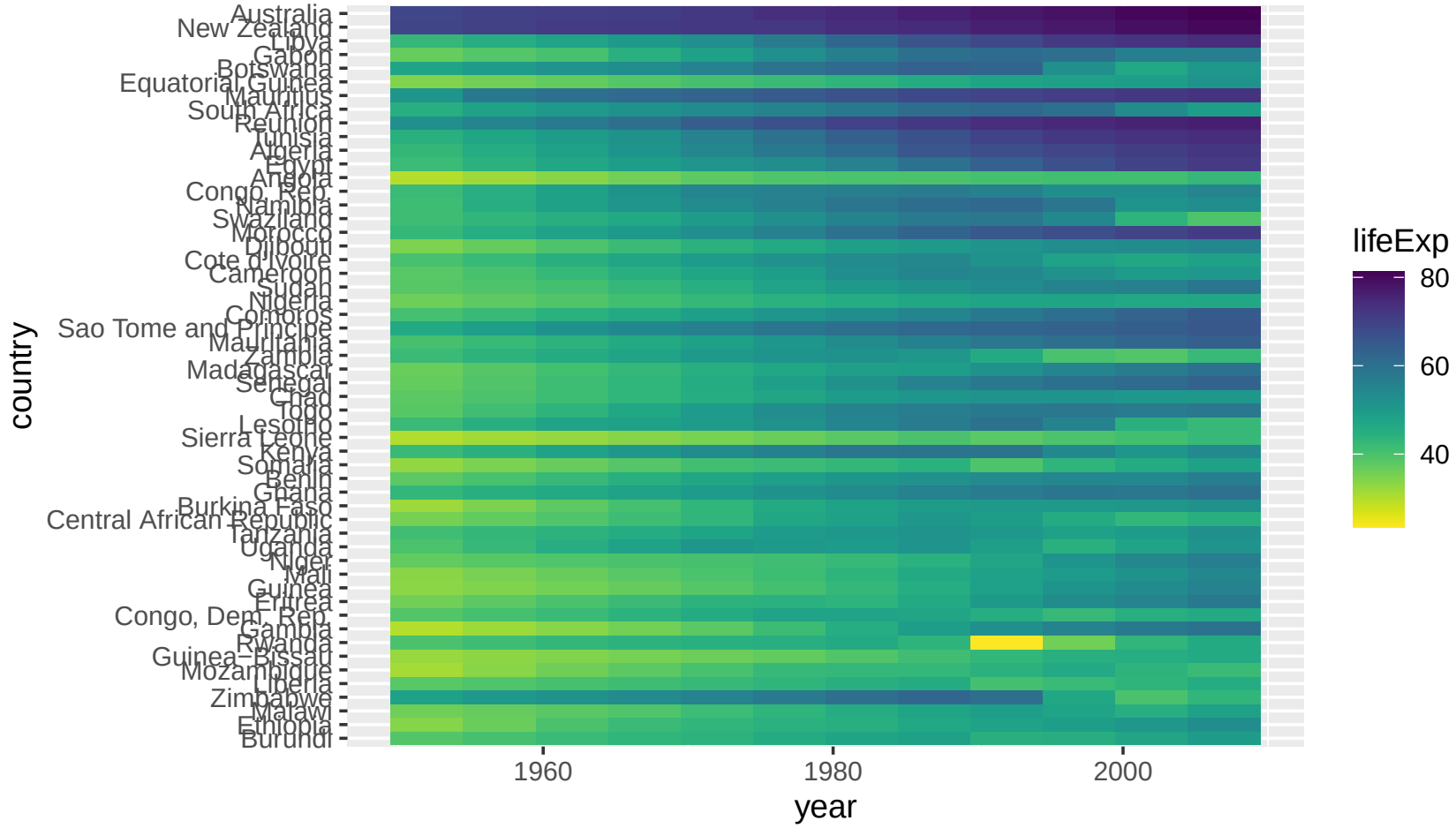
And from the `viridis` package.

```
library(viridis)
ggplot(gapminder, aes(x=year, y=country, fill=lifeExp)) +
    geom_tile() +
    scale_fill_viridis()
```



Trying out different color scales

```
ggplot(gapminder, aes(x=year, y=country, fill=lifeExp)) +
    geom_tile() +
    scale_fill_viridis(direction=-1)
```



Breakout rooms

Work with your group on the following:

- as a group, finish the note-catcher for last week, on recreating the improving the C02 emissions figure.
- start to look for an article for Lab 2 (see assignment on Moodle). You must complete this assignment on your own, but it's recommended to find an article that you can work in parallel with 1-2 other people on.