Data Visualization Case Study: Trends in World Health and Economics

Jasmine Kobayashi

09/19/2022

In this section we will demonstrate how relatively simple `ggplot` code can create insightful and aesthetically pleasing plots that help us better understand trends in world health and economics. We later augment the code somewhat to perfect the plots and describe some general principles for data visualization.

Example 1: Life Expectancy and Fertility Rates

Hans Rosling was the co-founder of the Gapminder Foundation, an organization dedicated to educating the public by using data to dispel common myths about the so-called developing world. The organization uses data to show how actual trends in health and economics contradict the narratives that emanate from sensationalist media coverage of catastrophes, tragedies and other unfortunate events. As stated on the Gapminder Foundation's website:

> Journalists and lobbyists tell dramatic stories. That's their job. They tell stories about extraordinary events and unusual people. The piles of dramatic stories pile up in peoples' minds into an over-dramatic worldview and strong negative stress feelings: "The world is getting worse!", "It's we vs. them!", "Other people are strange!", "The population just keeps growing!" and "Nobody cares!" Hans Rosling conveyed actual data-based trends in a dramatic way of his own, using effective data visualization. This section is based on two talks that exemplify this approach to education: New Insights on Poverty and The Best Stats You've Ever Seen. Specifically, in this section, we set out to answer the following two questions using data:

1. Is it a fair characterization of today's world to say it is divided into western rich nations and the developing world in Africa, Asia and Latin America?
2. Has income inequality across countries worsened during the last 40 years?

To answer these questions we will be using the `gapminder` dataset provided in `dslabs`. This dataset was created using a number of spreadsheets available from the Gapminder Foundation. You can access the table like this:

```
library(dslabs)
data(gapminder)
head(gapminder)
```

```
##               country year infant_mortality life_expectancy fertility
## 1             Albania 1960           115.40           62.87      6.19
## 2             Algeria 1960           148.20           47.50      7.65
## 3              Angola 1960           208.00           35.98      7.32
## 4 Antigua and Barbuda 1960               NA           62.97      4.43
## 5           Argentina 1960            59.87           65.39      3.11
## 6             Armenia 1960               NA           66.86      4.55
##   population          gdp continent          region
## 1    1636054           NA    Europe Southern Europe
## 2   11124892  13828152297    Africa Northern Africa
## 3    5270844           NA    Africa   Middle Africa
## 4      54681           NA  Americas       Caribbean
## 5   20619075 108322326649  Americas   South America
## 6    1867396           NA      Asia    Western Asia
```

Child Mortality Rates As done in the *New Insights on Poverty* video, we start by testing our knowledge regarding differences in child mortality across different countries.

For each of the six pairs of countries below, which country do you think had the highest child mortality in 2015? Which pairs do you think are most similar?

1. Sri Lanka or Turkey
2. Poland or South Korea
3. Malaysia or Russia
4. Pakistan or Vietnam
5. Thailand or South Africa

When answering these questions without data, the non-European countries are typically picked as having higher mortality rates: Sri Lanka over Turkey, South Korea over Poland, and Malaysia over Russia. It is also common to assume that countries considered to be part of the developing world, Pakistan, Vietnam, Thailand and South Africa, have similarly high mortality rates.

To answer these questions **with data** we can use dplyr. For example, for the first comparison we see that Turkey has the higher rate.

```
library(tidyverse)
gapminder %>% filter(year==2015 & country %in% c("Sri Lanka", "Turkey")) %>%
  select(country, infant_mortality)
```

```
##     country infant_mortality
## 1 Sri Lanka              8.4
## 2    Turkey             11.6
```

We can use this code on all comparisons and find the following:

Country_1	Infant_Mortality_1	Country_2	Infant_Mortality_2
Sri Lanka	8.4	Turkey	11.6
Poland	4.5	South Korea	2.9
Malaysia	6.0	Russia	8.2
Pakistan	65.8	Vietnam	17.3
Thailand	10.5	South Africa	33.6

We see that the European countries have higher rates: Poland has a higher rate than South Korea, and Russia has a higher rate than Malaysia. We also see that Pakistan has a much higher rate than Vietnam and South Africa a much higher rate than Thailand. It turns out that most people do worse if they are guessing, which implies we are more than ignorant, we are misinformed.

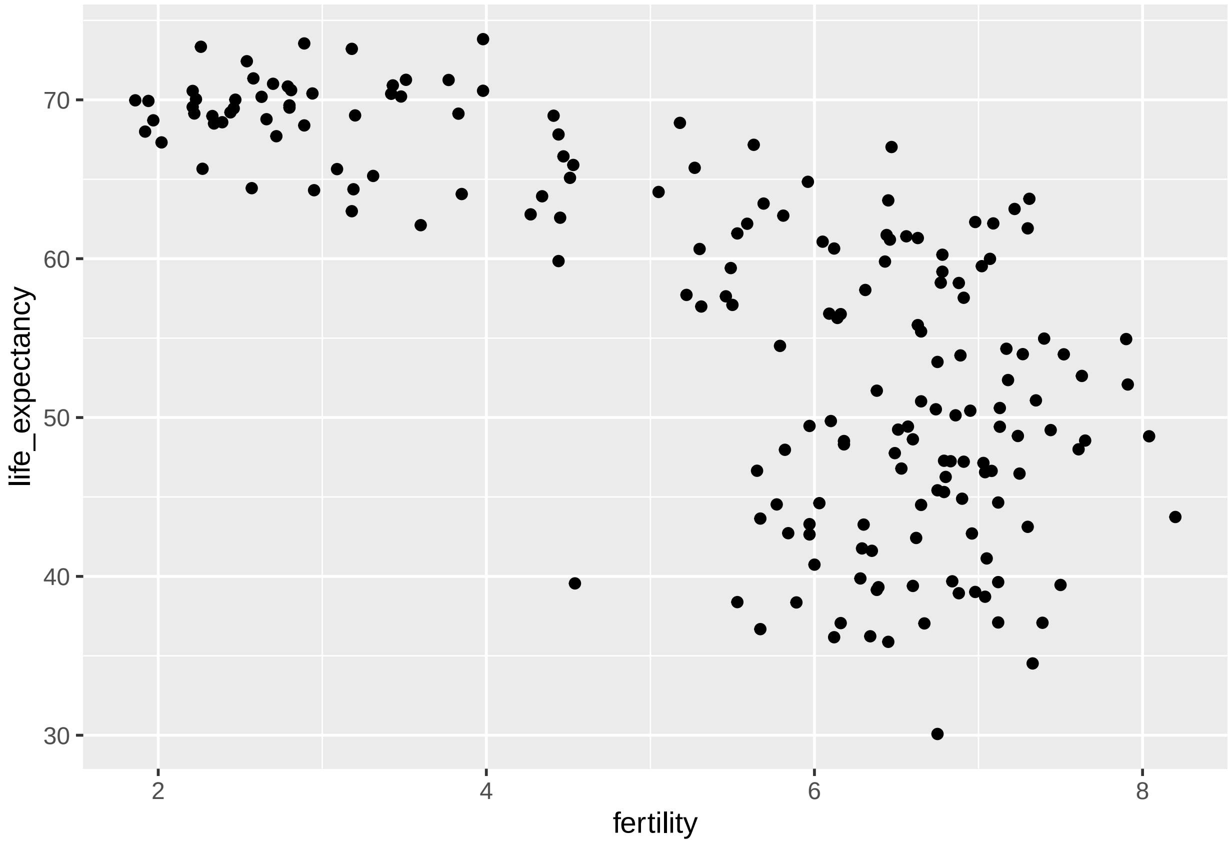
Life expectancy and fertility rates

The reason for this stems from the preconceived notion that the world is divided into two groups: the western world (Western Europe and North America), characterized by long life spans and small families, versus the developing world (Africa, Asia, and Latin America) characterized by short life spans and and large families. But, does the data support this dichotomous view of two groups?

The necessary data to answer this question is also available in our `gapminder` table. Using our newly learned data visualization skills we will be able to answer this question.

The first plot we make to see what data have to say about this world view is a scatter plot of life expectancy versus fertility rates (average number of children per woman). We will start by looking at data from about 50 years ago, when perhaps this view was cemented in our minds.

```
gapminder %>% filter(year==1962) %>% ggplot(aes(fertility, life_expectancy)) + geom_point()
```

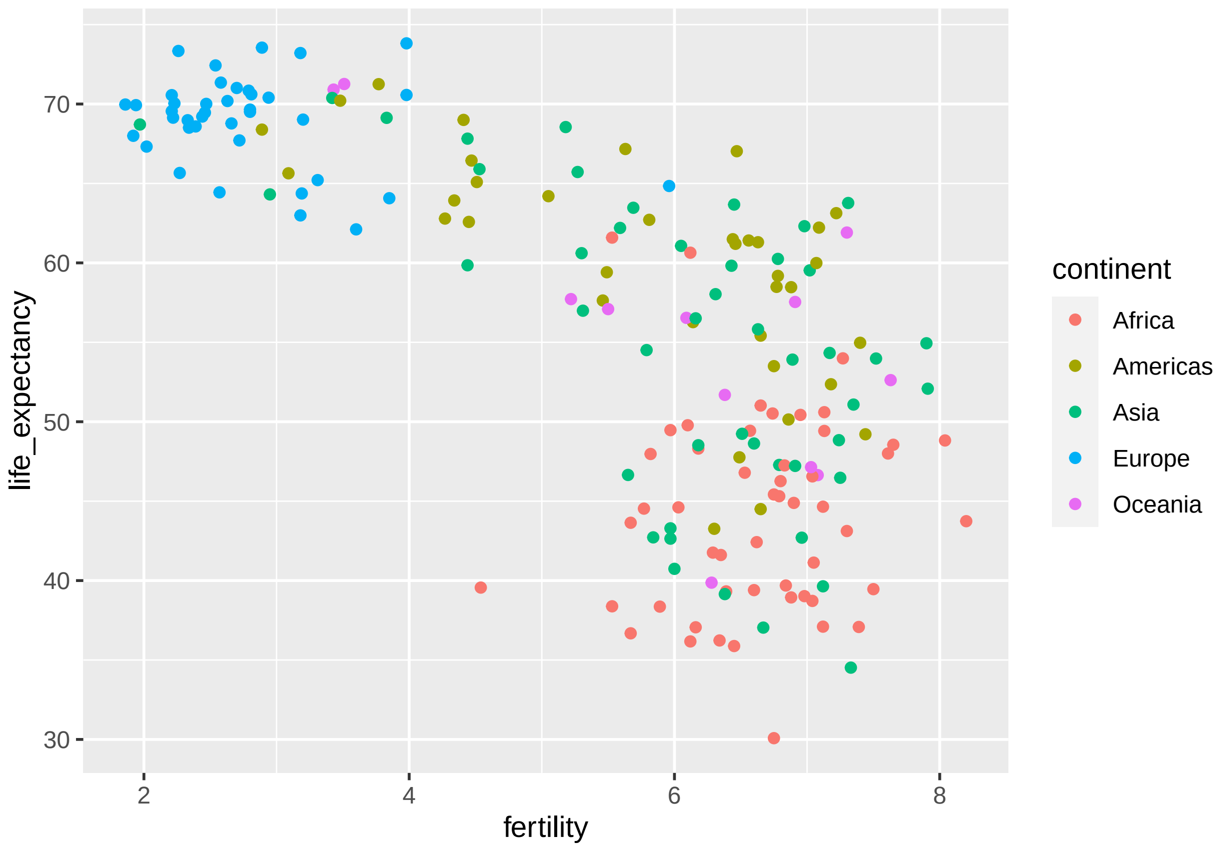


Most points fall into two distinct categories:

1. Life expectancy around 70 years and 3 or less children per family
2. Life expediencies lower than 65 years and with more than 5 children per family.

To confirm that indeed these countries are from the regions we expect, we can use color to represent continent.

```
gapminder %>% filter(year==1962) %>% ggplot(aes(fertility, life_expectancy, color=continent)) + geom_po
```



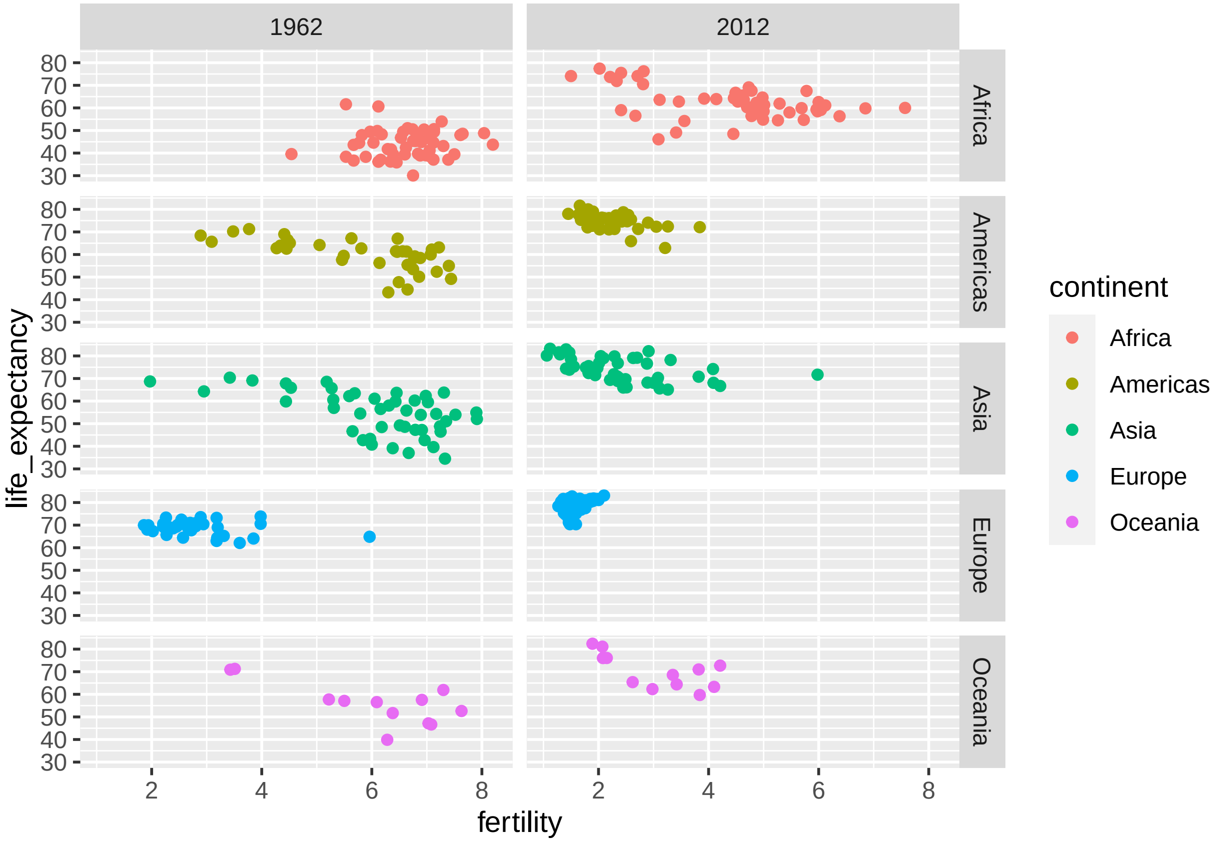
So in 1962, "the west versus developing world" view was grounded in some reality. But is this still the case 50 years later?

Faceting

We could easily plot the 2012 data in the same way we did for 1962. But to compare, side by side plots are preferable. In `ggplot` we can achieve this by *faceting variables*: we stratify the data by some variable and make the same plot for each stratum.

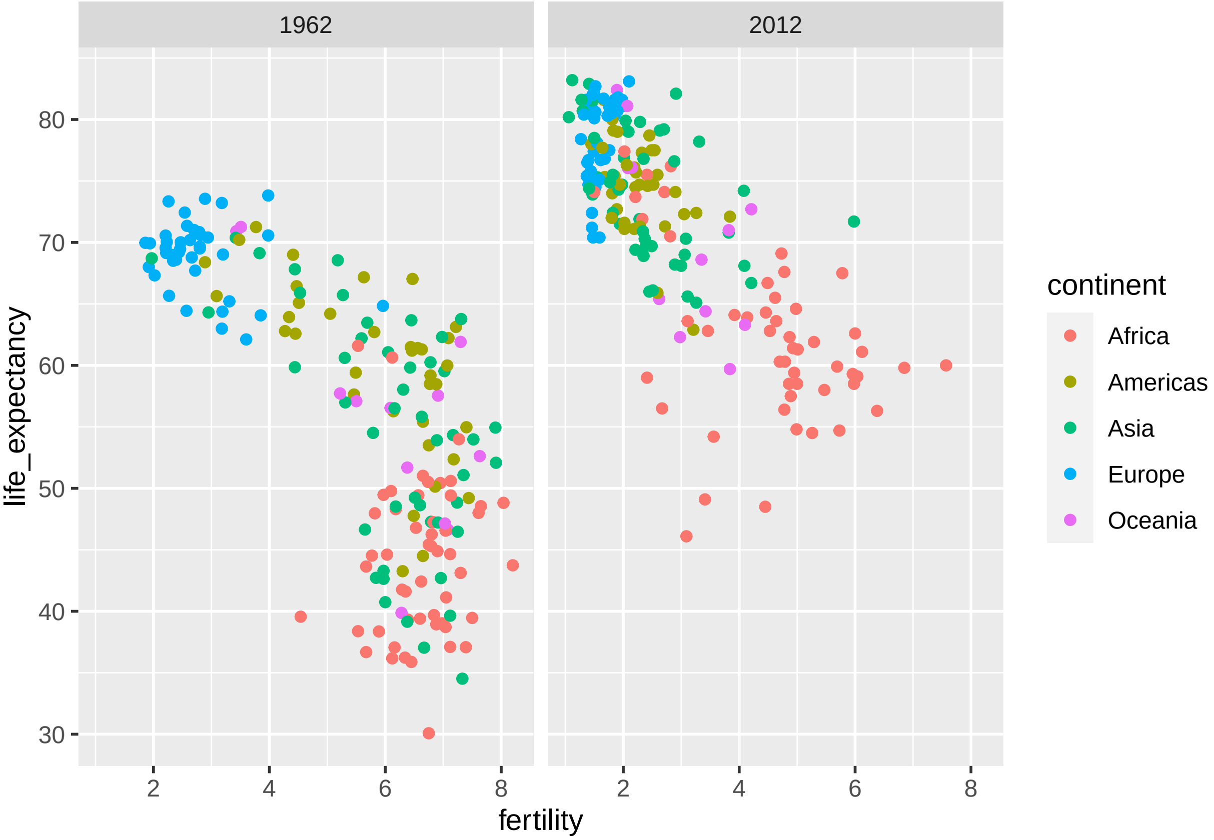
To achieve faceting we add a layer with the function `facet_grid`, which automatically separates the plots. This function lets you facet by up to two variables using columns to represent one variable and rows to represent the other. The function expects the row and column variables separated by a `~`. Here is an example of a scatter plot with `facet_grid` added as the last layer:

```
gapminder %>% filter(year %in% c(1962,2012)) %>% ggplot(aes(fertility, life_expectancy, color=continent
  geom_point() +
  facet_grid(continent ~ year)
```



We see a plot for each continent/year pair. However, this is just an example, and more than what we want, which is simply to compare 1962 and 2012. In this case, there is just one variable and we use . to let facet know that we are not using one of the variables:

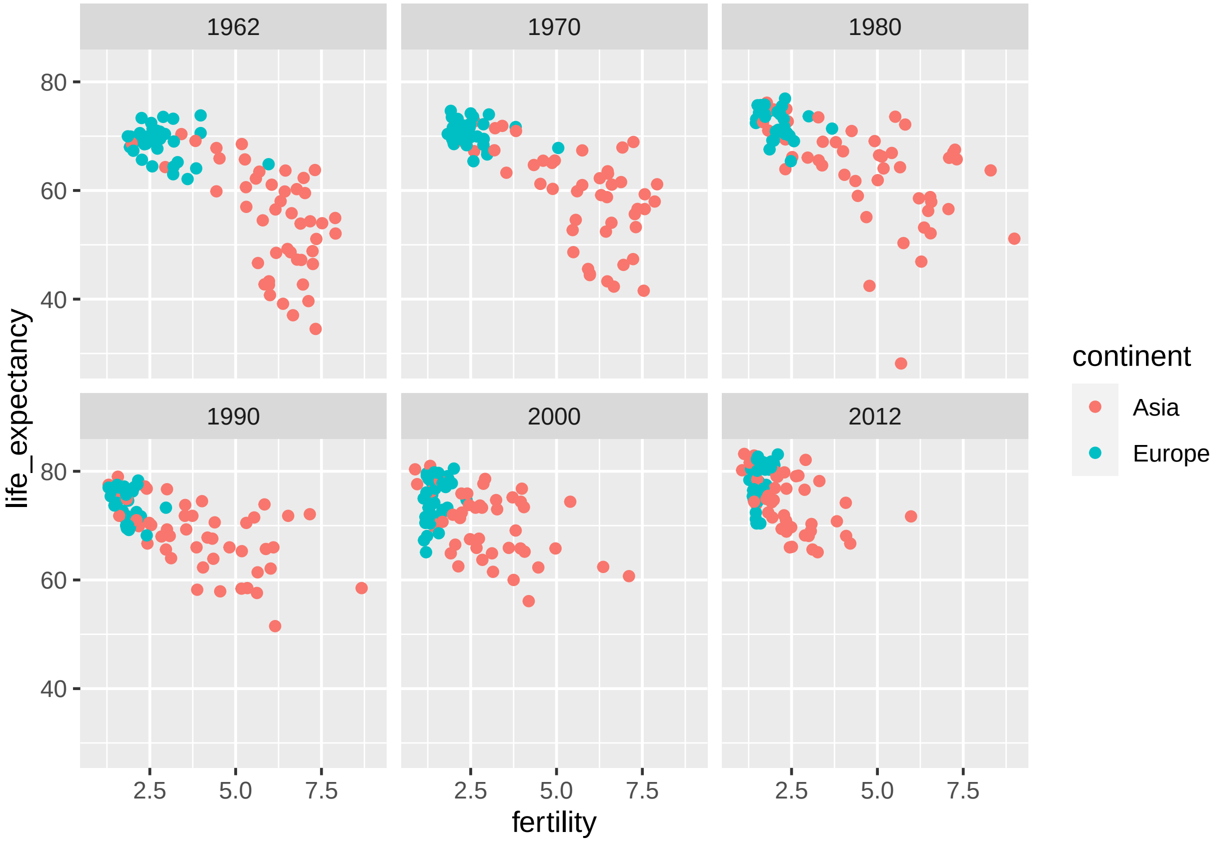
```
gapminder %>% filter(year %in% c(1962,2012)) %>% ggplot(aes(fertility, life_expectancy, color=continent)
  geom_point() +
  facet_grid(.~ year)
```



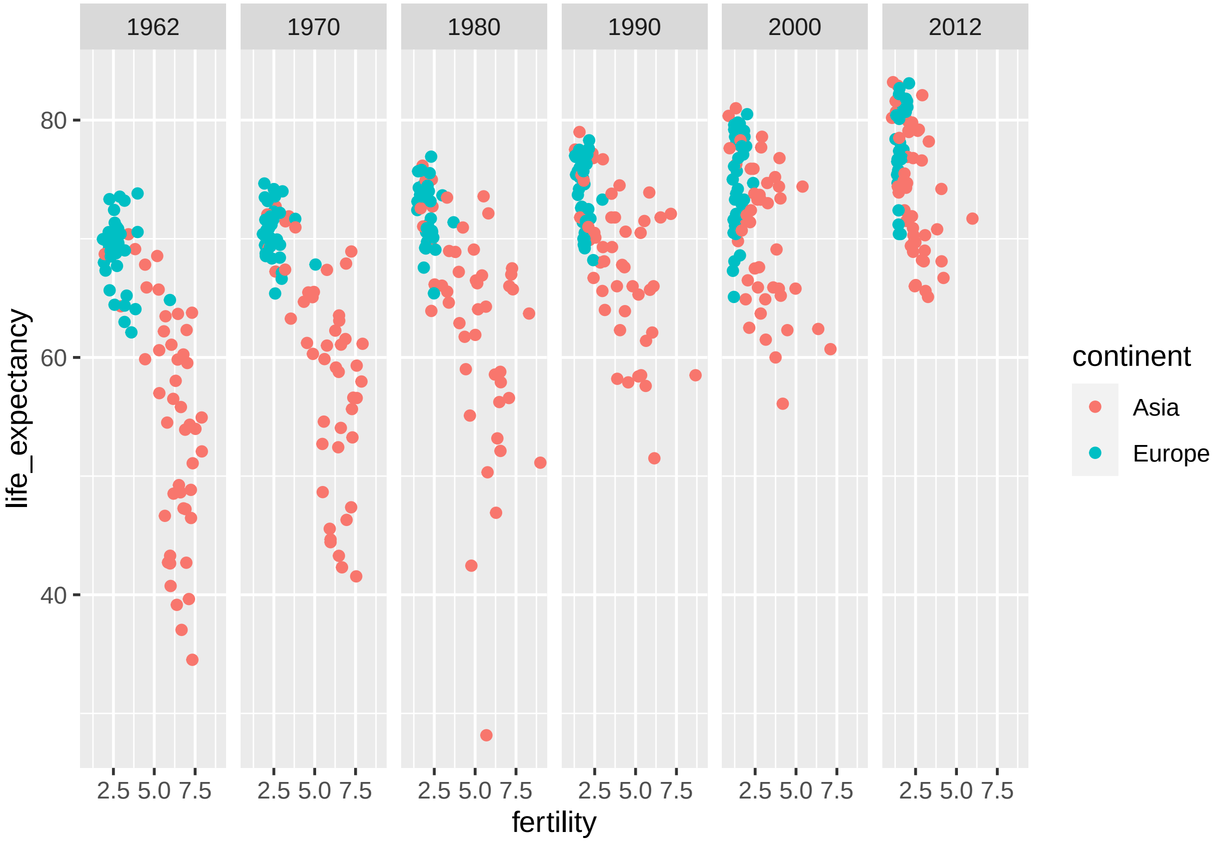
This plot clearly shows that the majority of countries have moved from the *developing world* cluster to the *western world* one. In 2012, the western versus developing world view no longer makes sense. This is particularly clear when comparing Europe to Asia, which includes several countries that have made great improvements.

`facet_wrap` To explore how this transformation happened through the years, we can make the plot for several years. For example we can add 1970, 1980, 1990, and 2000. If we do this, we will not want all the plots on the same row, the default behavior of `facet_grid`, as they will become too thin to show the data. Instead we will want to use multiple rows and columns. The function `facet_wrap` permits us to do this, as it automatically wraps the series of plots so that each display has viewable dimensions:

```r
years <- c(1962,1970,1980,1990,2000,2012)
continents <- c("Europe","Asia")
gapminder %>% filter(year %in% years & continent %in% continents) %>%
  ggplot(aes(fertility,life_expectancy,color=continent)) +
  geom_point() +
  facet_wrap(. ~ year)
```



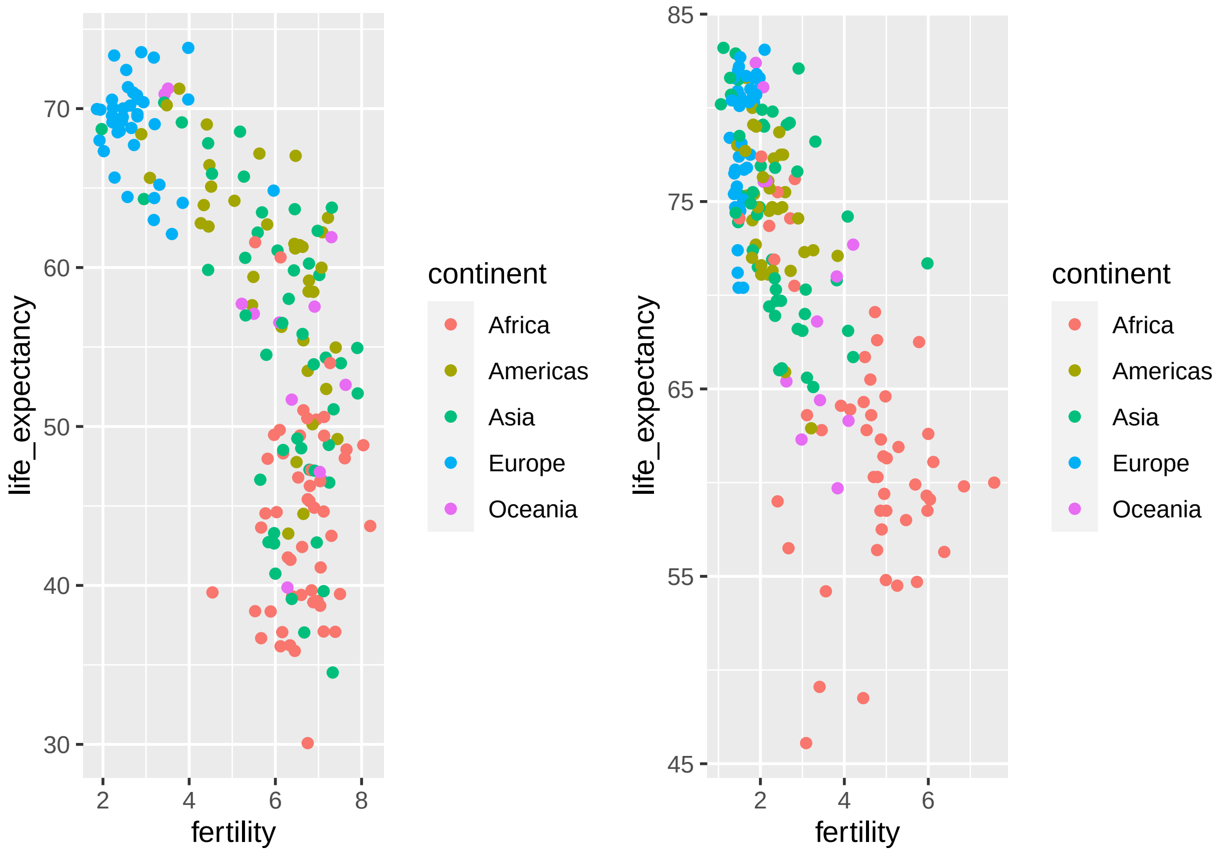
```
# testing use of facet_grid
years <- c(1962,1970,1980,1990,2000,2012)
continents <- c("Europe","Asia")
gapminder %>% filter(year %in% years & continent %in% continents) %>%
  ggplot(aes(fertility,life_expectancy,color=continent)) +
  geom_point() +
  facet_grid(. ~ year)
```



This plot clearly shows how most Asian countries have improved at a much faster rate than European ones.

Fixed scales for better comparisons

Note that the default choice of the range of the axes is an important one. When not using `facet`, this range is determined by the the data shown in the plot. When using `facet`, this range is determined by the data shown in all plots and therefore kept fixed across plots. This makes comparisons across plots much easier. For example, in the plot above we see that life expectancy has increased and the fertility has decreased across most countries. We see this because the cloud of points moves. This is not the case if we don't adjust the scales:



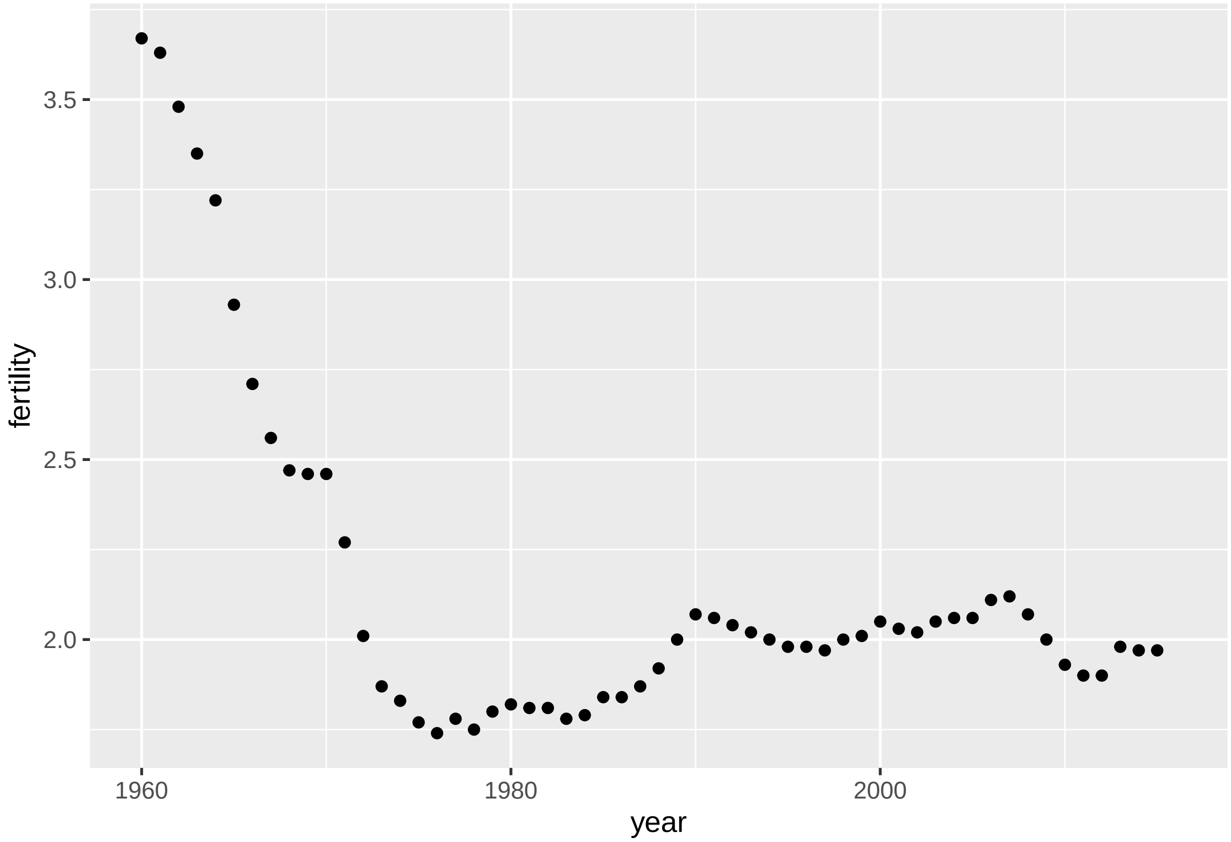
In the plot above we have to pay special attention to the range to notice that the plot on the right has larger life expectancy.

Time Series Plots

The visualizations above effectively illustrate that data no longer support the western versus developing world view. Once we see these plots new questions emerge. For example, which countries are improving more, which ones less? Was the improvement constant during the last 50 years or was there more accelerated improvement during certain periods? For a closer look that may help answer these questions, we introduce *time series plots*.

Time series plots have time on the x-axis and an outcome or measurement of interest on the y-axis. For example, here is a trend plot for the United States fertility rates:

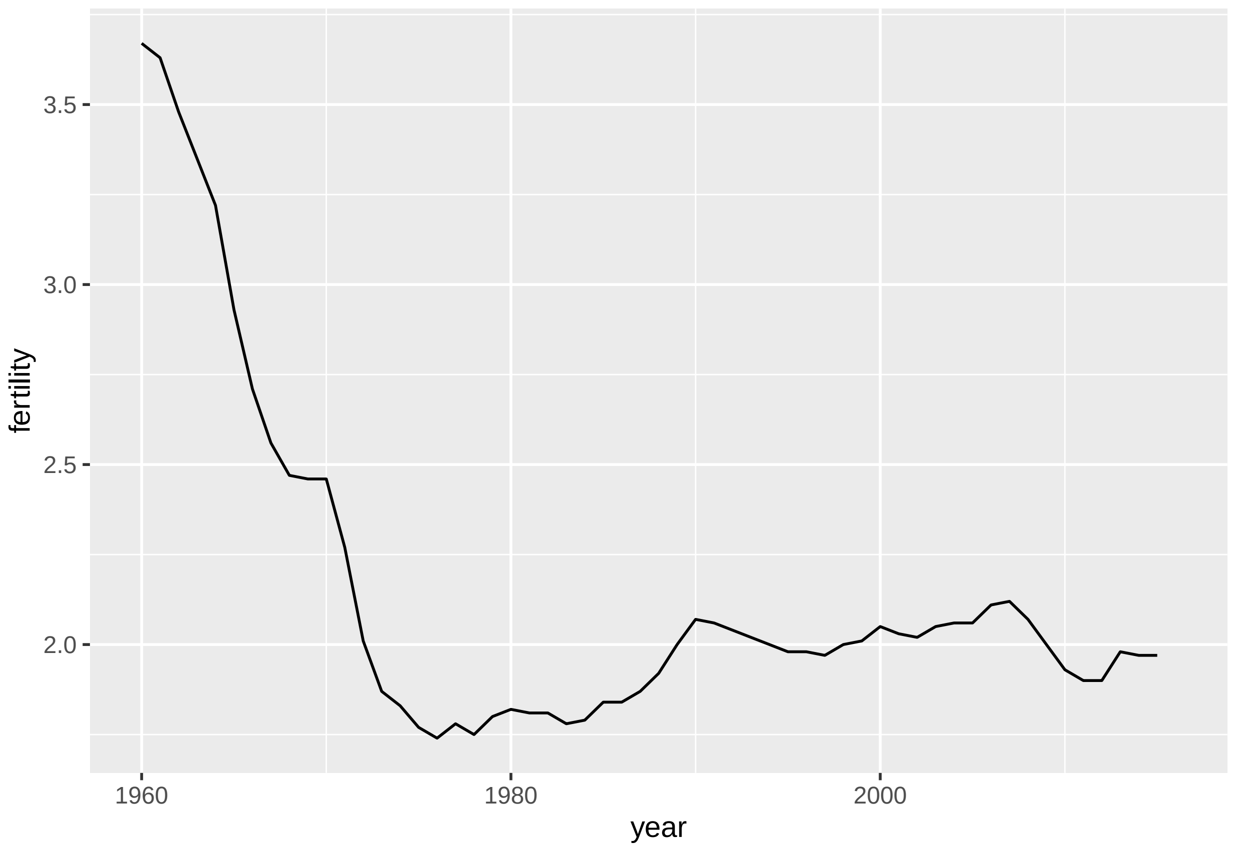
```
gapminder %>% filter(country == "United States") %>%
  ggplot(aes(year,fertility)) +
  geom_point()
```



We see that the trend is not linear at all. Instead we see a sharp drop during the 60s and 70s to below 2. Then the trend comes come back to 2 and stabilizes during the 90s.

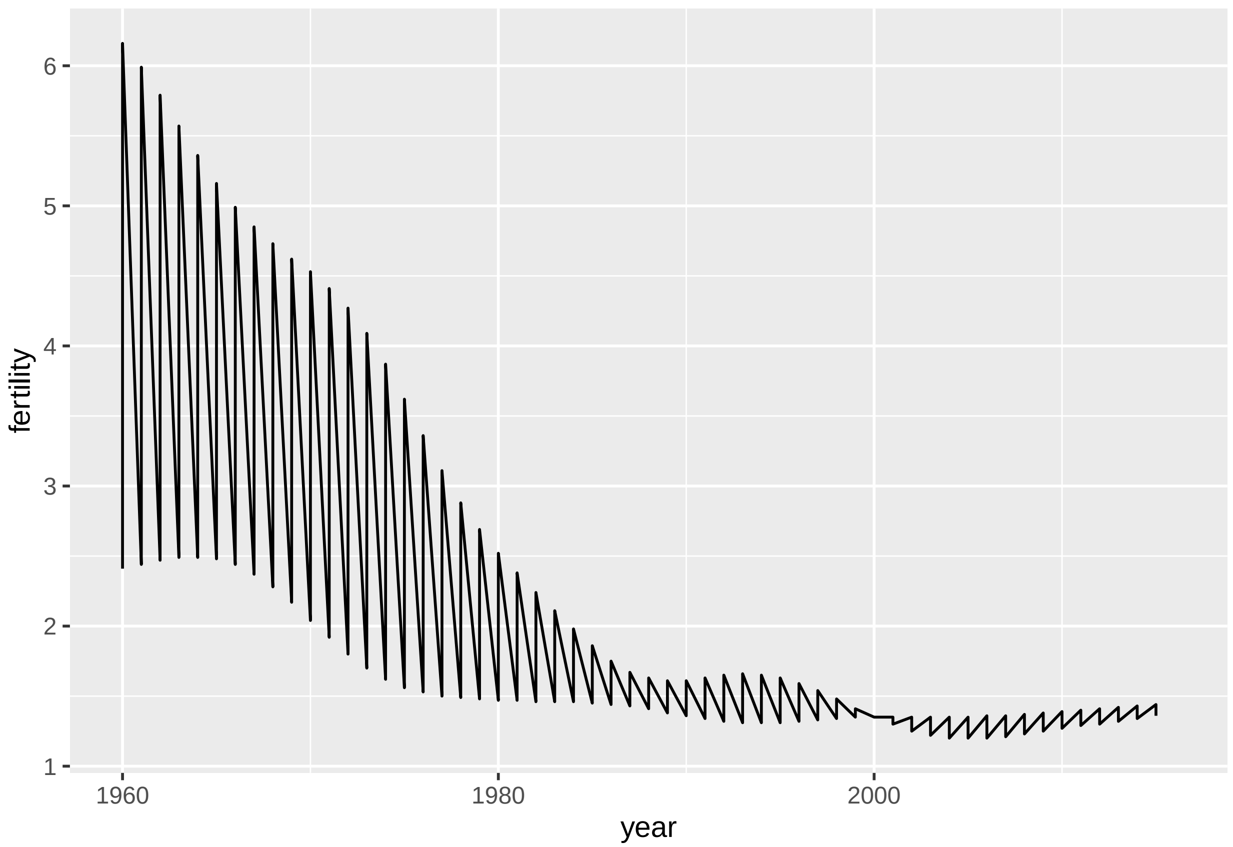
When the points are regularly and densely spaced, as they are here, we create curves by joining the points with lines, to convey that these data are from a single country. To do this we use the `geom_line` function instead of `geom_point`.

```
gapminder %>% filter(country == "United States") %>%
  ggplot(aes(year,fertility)) +
  geom_line()
```



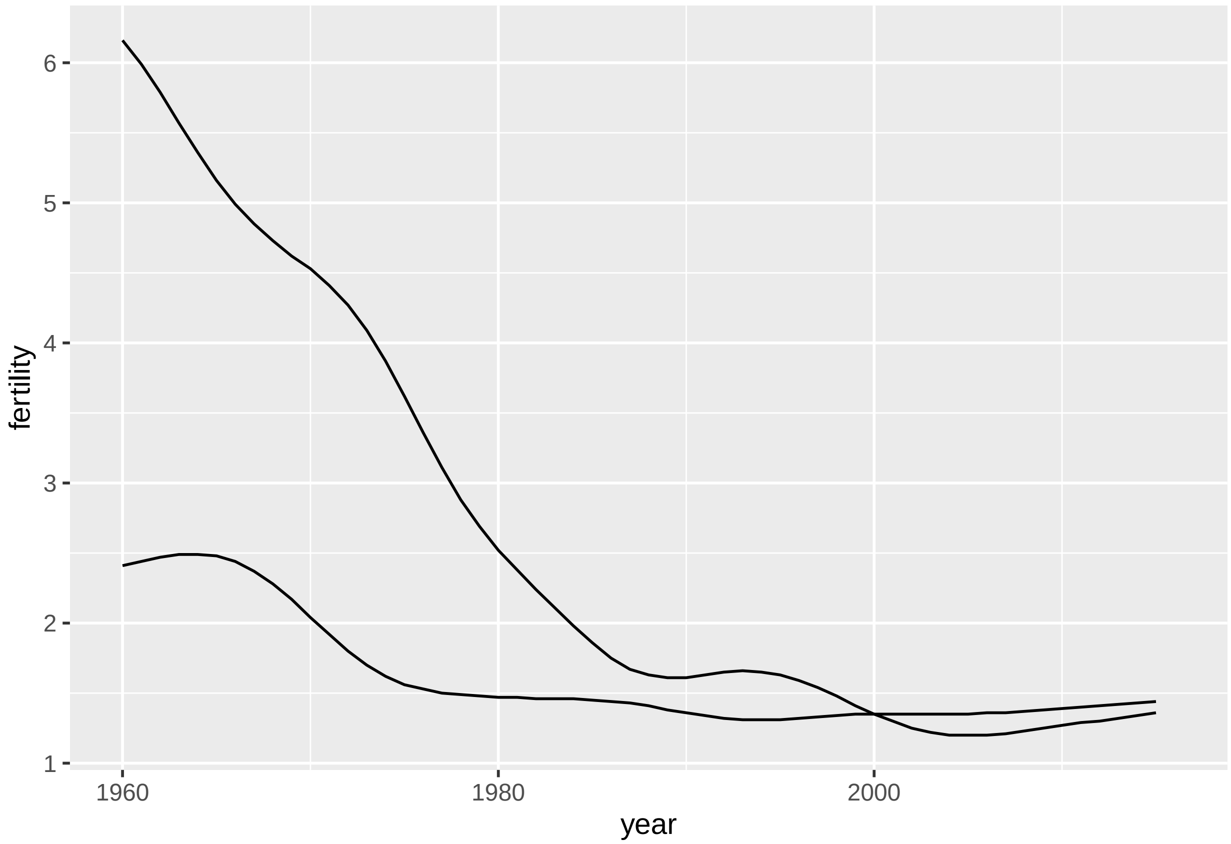
This is particularly helpful when we look at two countries. If we subset the data to include two countries, one from Europe and one from Asia, then copy the code above:

```
countries <- c("South Korea", "Germany")
gapminder %>% filter(country %in% countries ) %>%
  ggplot(aes(year,fertility)) +
  geom_line()
```



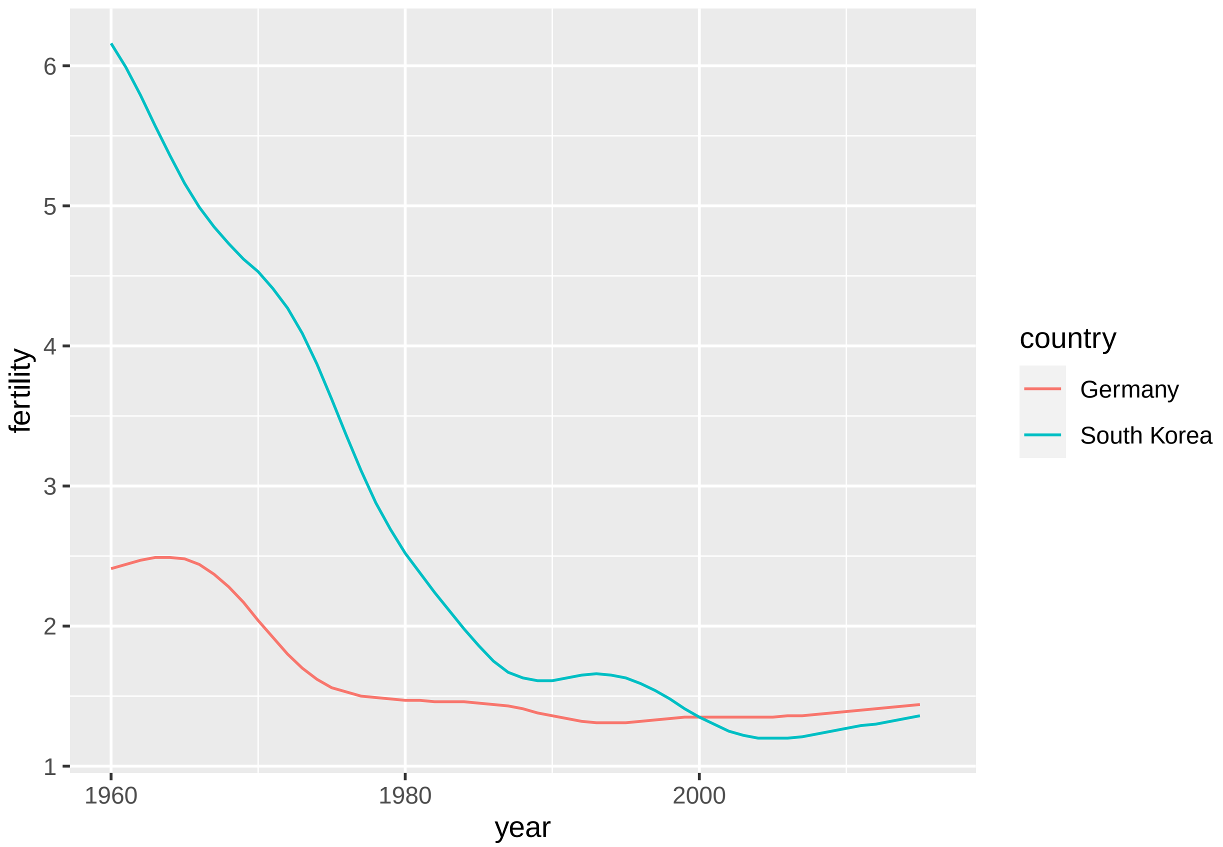
Note that this is **not** the plot that we want. Rather than a line for each country, the points for both countries are joined. This is actually expected since we have not told `ggplot` anything about wanting two separate lines. To let `ggplot` know that there are two curves that need to be made separately, we assign each point to a `group`, one for each country:

```r
countries <- c("South Korea", "Germany")
gapminder %>% filter(country %in% countries ) %>%
  ggplot(aes(year,fertility, group = country)) +
  geom_line()
```



But which line goes with which country? We can assign colors to make this distinction. A useful side-effect of using the `color` argument to assign different colors to the different countries is that the data is automatically grouped:

```
countries <- c("South Korea", "Germany")
gapminder %>% filter(country %in% countries ) %>%
  ggplot(aes(year,fertility, color = country)) +
  geom_line()
```



The plot clearly shows how South Korea's fertility rate dropped drastically during the 60s and 70s and by 1990 had a similar rate to Germany.

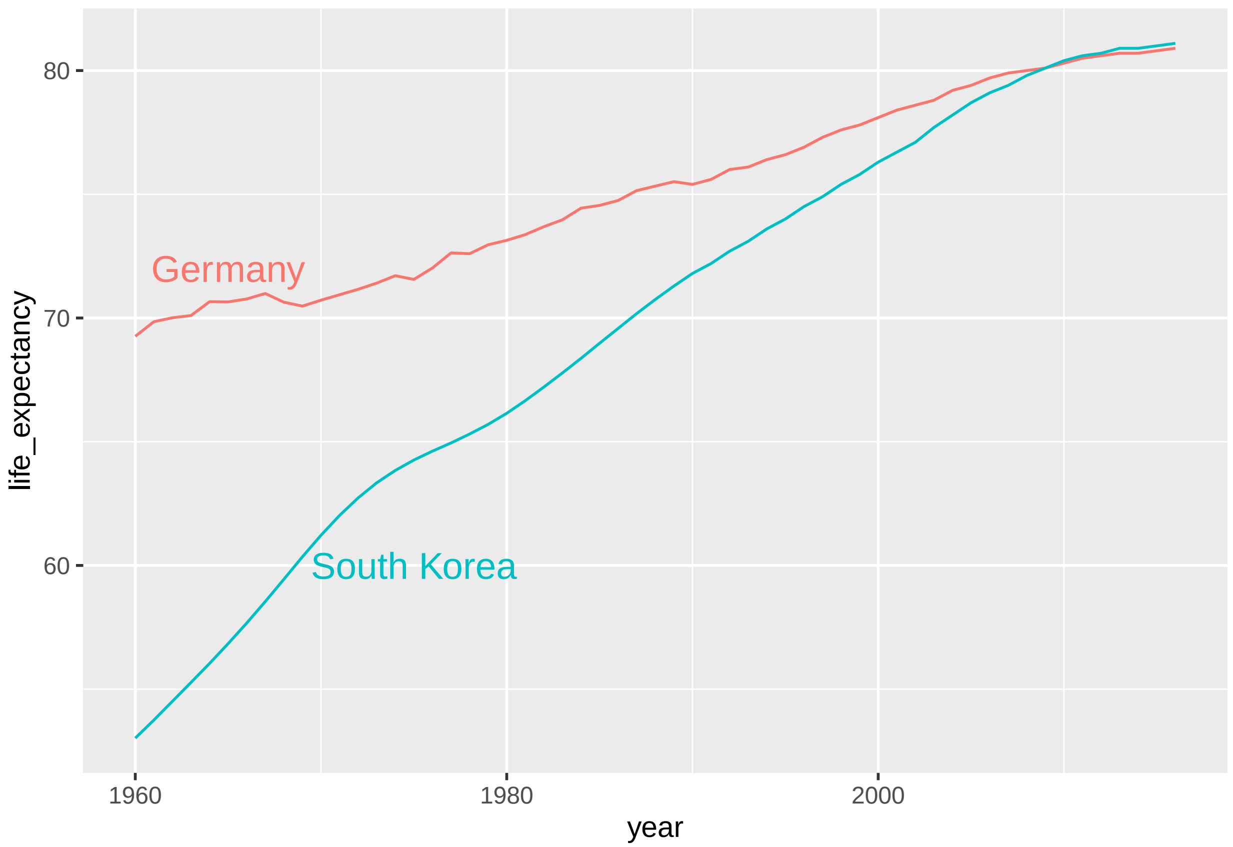
We prefer labels over legends (this will be on the quiz) For trend plots we recommend labeling the lines rather than using legends as the viewer can quickly see which line is which country. This suggestion actually applies to most plots: **labeling is usually preferred over legends.**

We demonstrate how we can do this using the life expectancy data. We define a data table with the label locations and then use a second mapping just for these labels:

```
labels <- data.frame(country = countries, x = c(1975, 1965), y = c(60, 72))
labels
```

```
##       country    x  y
## 1 South Korea 1975 60
## 2     Germany 1965 72
```

```
gapminder %>% filter(country %in% countries) %>%
  ggplot(aes(year, life_expectancy, color = country)) +
  geom_line() +
  geom_text(data = labels, aes(x, y, label = country), size = 5) +
  theme(legend.position = "none")
```



The plot clearly shows how an improvement in life expectancy followed the drops in fertility rates. While in 1960 Germans lived more than 15 years more South Koreans, by 2010 the gap is completely closed. It exemplifies the improvement that many non-western countries have achieved in the last 50 years.

Example 2: Income Distribution

Another commonly held notion is that wealth distribution across the world has become worse during the last several decades.

When general audiences are asked if poor countries have become poorer and rich countries become richer, the majority answer yes. By using stratification, histograms, smooth densities, and boxplots we will be able to understand if this is in fact the case. We will also learn how transformations can sometimes help provide more informative summaries and plots.

Transformations

The `gapminder` data table includes a column with the countries gross domestic product (GDP). GDP measures the market value of goods and services produced by a country in a year. The GDP per person is often used as a rough summary of how rich a country is. Here we divide this quantity by 365 to obtain the more interpretable measure *dollars per day*. Using current US dollars as a unit, a person surviving on an income of less than $2 a day is defined to be living in *absolute poverty*. We add this variable to the data table:

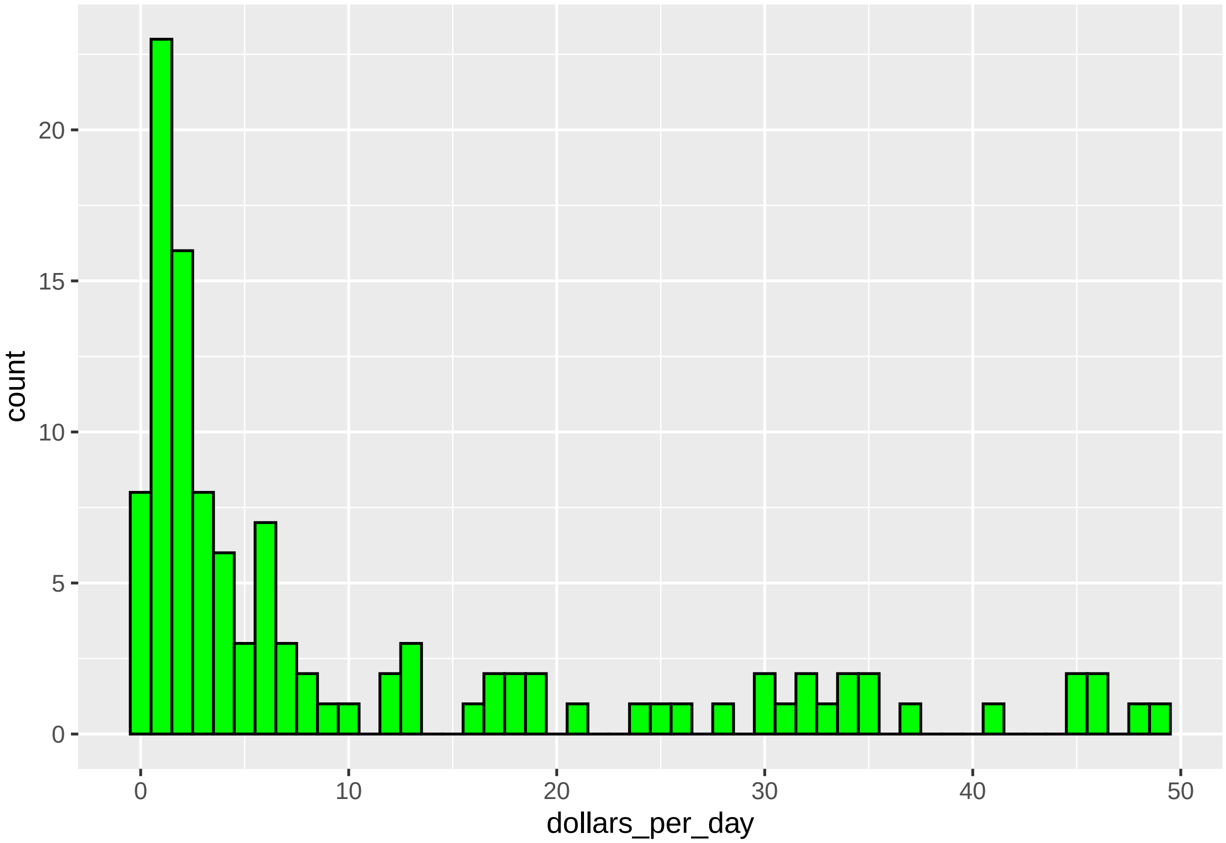
```
gapminder <- gapminder %>% mutate(dollars_per_day = gdp/population/365)
```

Note that the GDP values are adjusted for inflation and represent current US dollars, so these values are meant to be comparable across the years. Also note that these are country averages and that within each country there is much variability. **All the graphs and insights described below relate to country averages and not to individuals.**

Country income distribution Here is a histogram of per day incomes from 1970:

```
past_year <- 1970

gapminder %>% filter(year == past_year & !is.na(gdp)) %>%
  ggplot(aes(dollars_per_day)) + geom_histogram(binwidth = 1, color = "black", fill = "green")
```



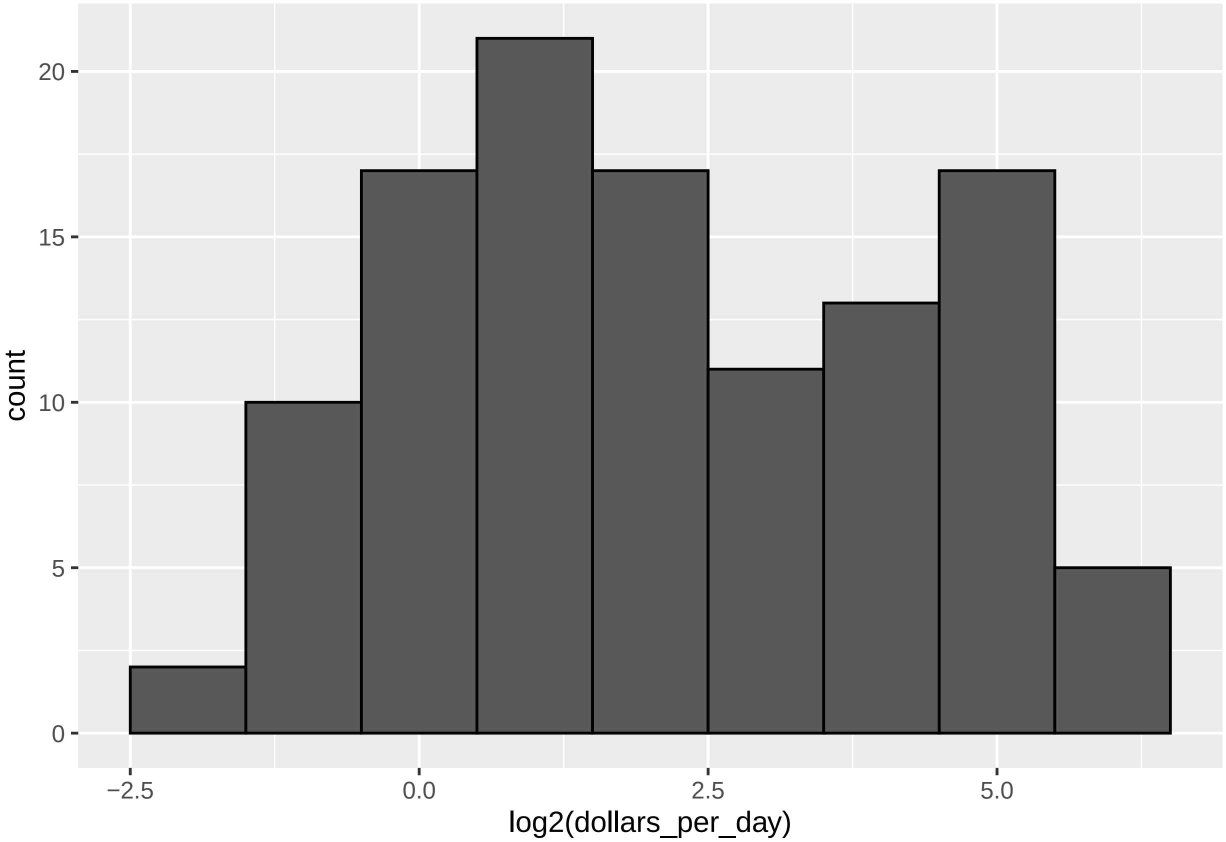
We use the `color = "black"` argument to draw a boundary and clearly distinguish the bins.

In this plot we see that for the majority of countries, averages are below $10 a day. However, the majority of the x-axis is dedicated to the 35 countries with averages above $10. So the plot is not very informative about countries with values below $10 a day.

It might be more informative to quickly be able to see how many countries have average daily incomes of about $1 (extremely poor), $2 (very poor), $4 (poor), $8 (middle), $16 (well off), $32 (rich), $64 (very rich) per day. These changes are multiplicative and log transformations change multiplicative changes into additive ones: when using base 2, a doubling of a value turns into an increase by 1.

Here is the distribution if we apply a log base 2 transformation:

```
gapminder %>%
  filter(year == past_year & !is.na(gdp)) %>%
  ggplot(aes(log2(dollars_per_day))) + geom_histogram(binwidth = 1, color = "black")
```



In a way this provides a *close up* of the mid to lower income countries.

Which base? In the case above we used base 2 in the log transformations. Other common choices are base e (the natural log) and base 10.

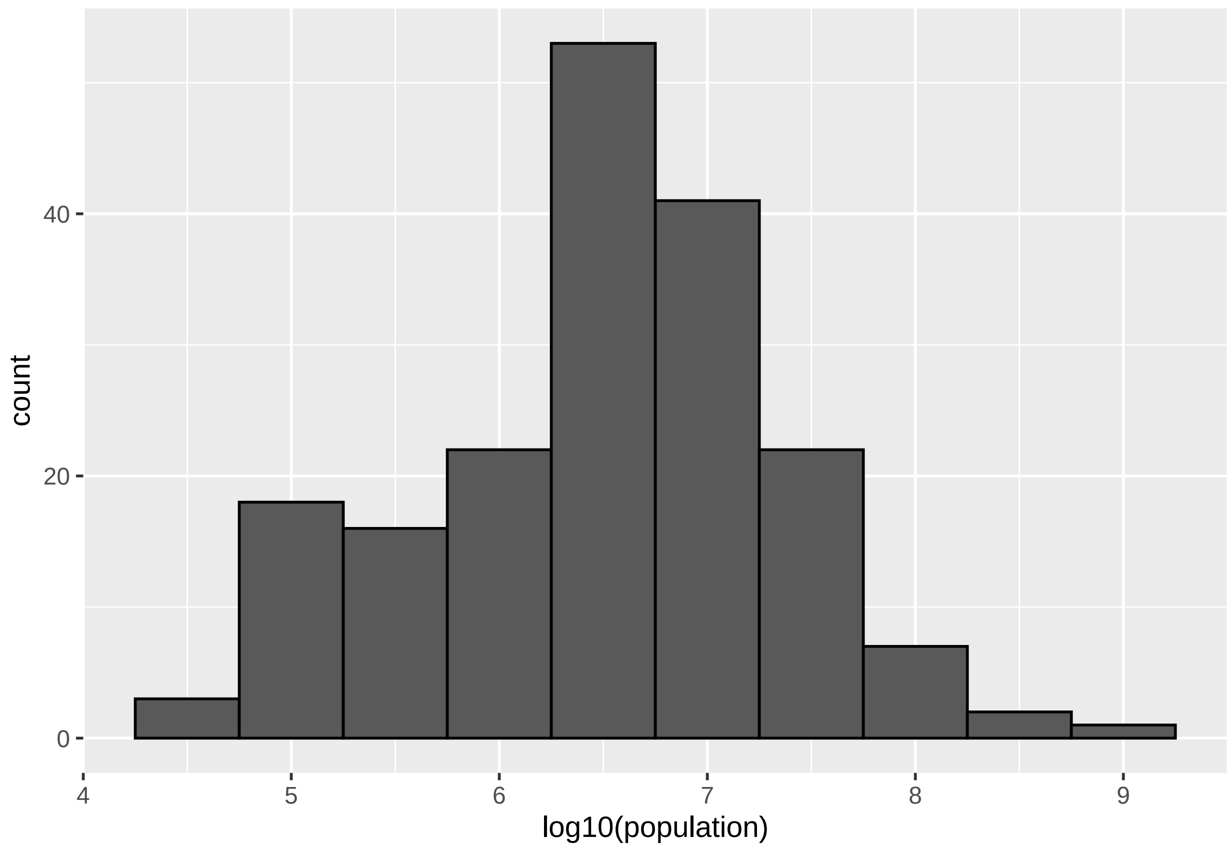
In general, we do not recommend using the natural log for data exploration and visualization. This is because while $2^2, 2^3, 2^4, \dots$ or $10^1, 10^2, \dots$ are easy to compute in our heads, the same is not true for $\mathrm{e}^2, \mathrm{e}^3, \dots$.

In the dollars per day example, we used base 2 instead of base 10 because the resulting range is easier to interpret. The range of the values being plotted is (0.3269426, 48.8852142).

In base 10 this turns into a range that includes very few integers: just 0 and 1. With base two, our range includes -2, -1, 0, 1, 2, 3, 4 and 5. It is easier to compute 2^x and 10^x when x is an integer and between -10 and 10, so we prefer to have more small integers on the scale. Another consequence of a limited range is that choosing the binwidth is more challenging. With log base 2, we know that a binwidth of 1 will translate to a bin with range x to $2x$.

As an example in which base 10 makes more sense consider population sizes. A log base 10 makes more sense since the range for these is about 1,000 to 10 billion. Here is the histogram of the transformed values:

```
gapminder %>% filter(year==past_year) %>% ggplot(aes(log10(population))) +
  geom_histogram(binwidth = .5, color = "black")
```



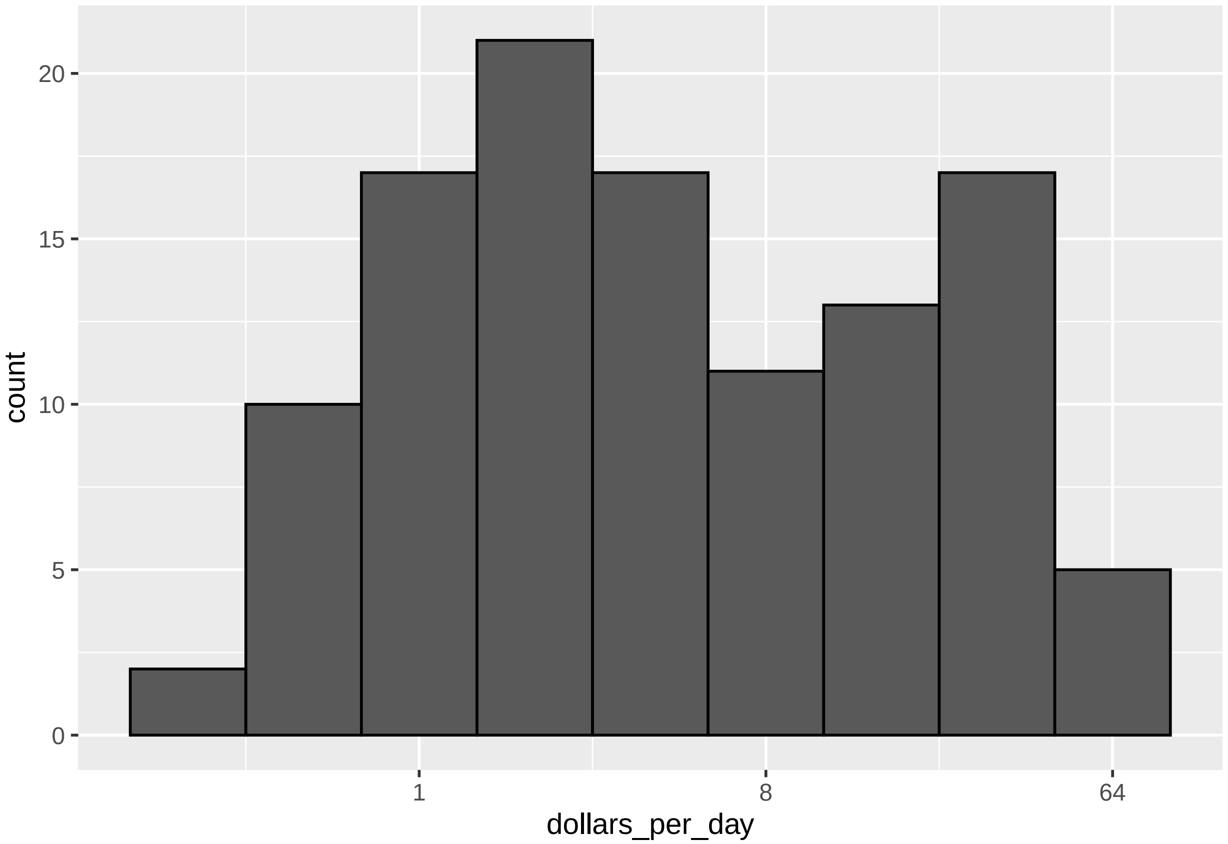
Here we quickly see that country populations range between ten thousand and ten billion.

Transform the values or the scale? There are two ways we can use log transformations in plots. We can log the values before plotting them or use log scales in the axes. Both approaches are useful and have different strengths. If we log the data we can more easily interpret intermediate values in the scale. For example, if we see

> —-1—-x—-2——–3—- for log transformed data we know that the value of x is about 1.5. If the scales are logged
>
> —-1—-x—-10——100— then, to determine `x`, we need to compute $10^{1.5}$, which is not easy to do in our heads. However, the advantage of showing logged scales is that the original values are displayed in the plot, which are easier to interpret. For example, we would see "32 dollars a day" instead of "5 log base 2 dollar a day".

As we learned earlier, if we want to scale the axis with logs we can use the `scale_x_continuous` function. So instead of logging the values first, we apply this layer:

```
gapminder %>% filter(year == past_year & !is.na(gdp)) %>% ggplot(aes(dollars_per_day)) +
  geom_histogram(binwidth = 1, color = "black") +
  scale_x_continuous(trans = "log2")
```



Note that the log base 10 transformation has it's own function: `scale_x_log10()`, but currently base 2 does not. Although we could easily define our own.

Note that there are other transformations available through the `trans` argument. As we learn later on, the square root (`sqrt`) transformation, for example, is useful when considering counts. The logistic transformation (`logit`) is useful when plotting proportions between 0 and 1. The `reverse` transformation is useful when we want smaller values to be on the right or on top.

Modes

The above plot show two "bumps". In statistics, these bumps are sometimes referred to as *modes*. The mode of a distribution is the value with the highest frequency. The mode of the normal distribution is the average. When a distribution, like the one above, doesn't monotonically decrease from the mode we call the locations where it goes up and down against local modes and say that the distribution has multiple modes.

The histogram above suggests that the 1970 country income distribution has two modes: one at about 2 dollars per day (1 in the log 2 scale) and another at about 32 dollars per day (5 in the log 2 scale). This *bimodality* is consistent with a dichotomous world made up of countries with average incomes less than $8 (3 in the log 2 scale) a day and countries above that.

Stratify and boxplot

The histogram showed us that the income distribution values show a dichotomy. However, the histogram does not show us if the two groups of countries are *west* versus the *developing* world.

To see distributions by geographical region we first stratify the data into regions, and then examine the distribution for each. Because of the number of regions,

```
levels(gapminder$region)
```

```
##  [1] "Australia and New Zealand" "Caribbean"
##  [3] "Central America"           "Central Asia"
##  [5] "Eastern Africa"            "Eastern Asia"
##  [7] "Eastern Europe"            "Melanesia"
##  [9] "Micronesia"                "Middle Africa"
## [11] "Northern Africa"           "Northern America"
## [13] "Northern Europe"           "Polynesia"
## [15] "South America"             "South-Eastern Asia"
## [17] "Southern Africa"           "Southern Asia"
## [19] "Southern Europe"           "Western Africa"
## [21] "Western Asia"              "Western Europe"
```

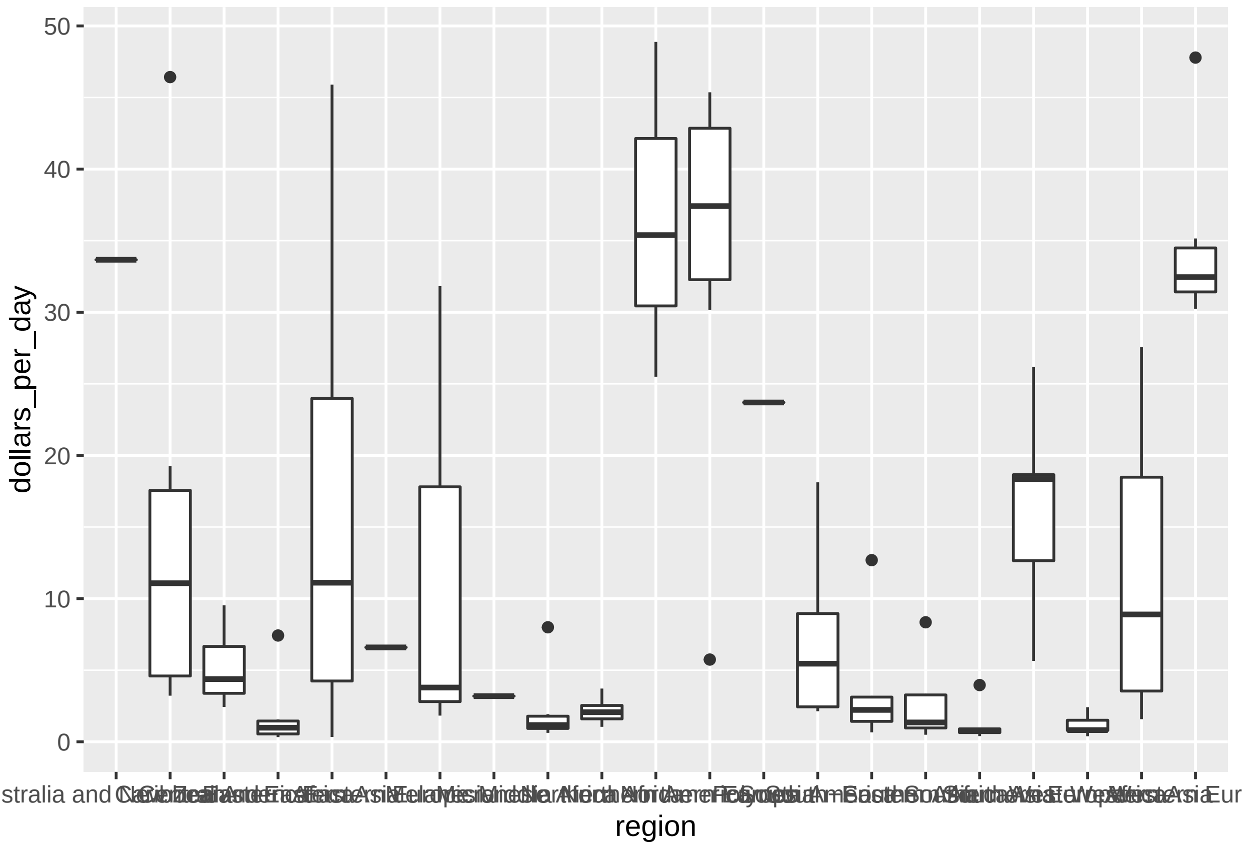
```
length(levels(gapminder$region))
```

```
## [1] 22
```

looking at histograms or smooth densities for each will not be useful. Instead, we can stack boxplots next to each other:

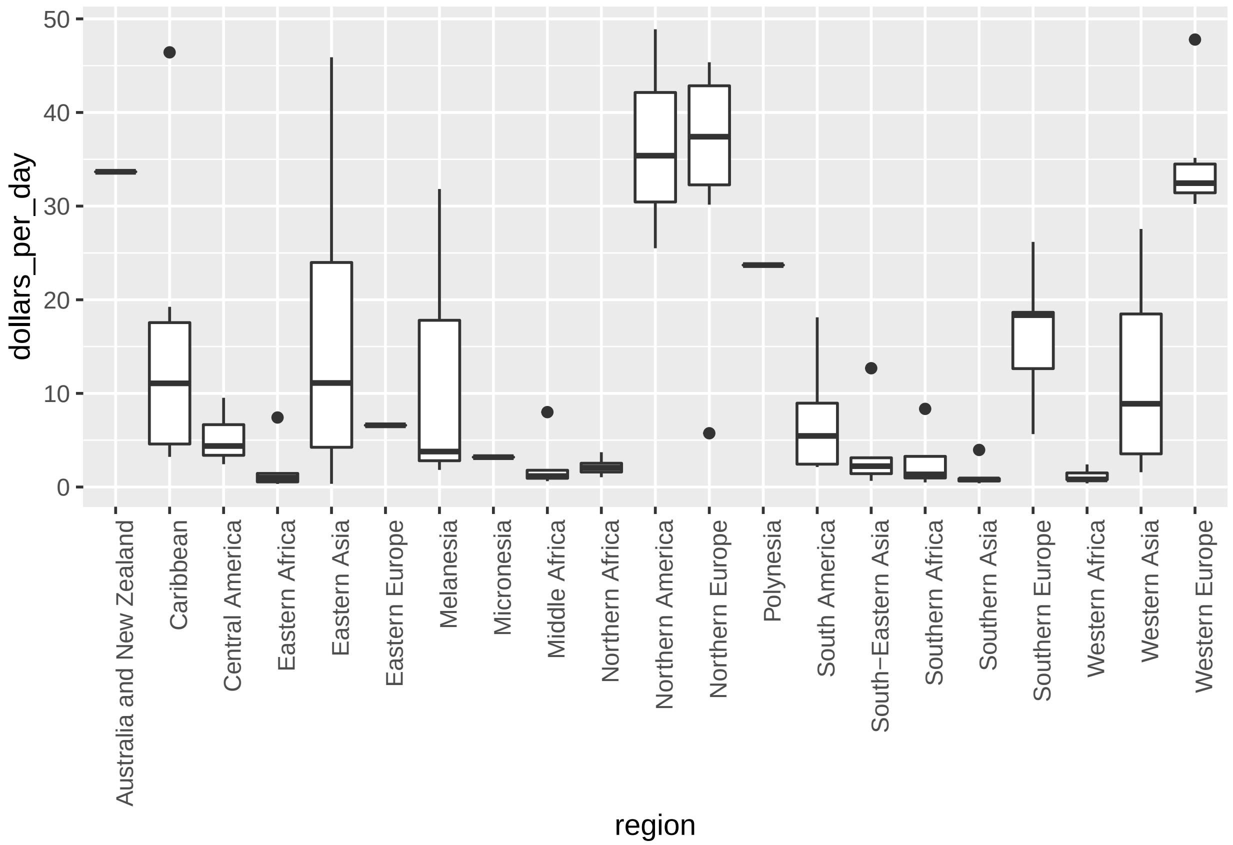
```
p <- gapminder %>% filter(year == past_year & !is.na(gdp)) %>%
  ggplot(aes(region, dollars_per_day))

p + geom_boxplot()
```



Note that we can't read the region names because the default gpplot behavior is to write the labels horizontally and, here, we run of of room. We can easily fix this by rotating the labels. Consulting the cheat sheet we find we can rotate the names by changing the `theme` through `element_text`. The `hjust=1` justifies the text so that it is next to the axis.

```
p + geom_boxplot() + theme(axis.text.x = element_text(angle = 90, hjust=1))
```



We can already see that there is indeed a west versus the rest dichotomy.

Do not order alphabetically There are a few more adjustments we can make to this plot help uncover this reality. First, it helps to order the regions in the boxplots from poor to rich rather than alphabetically. This can be achieved using the `reorder` function. This function let's us change the order of the levels of a factor variable based on a summary computed on a numeric vector. A character vector gets coerced into a factor. Here is an example. Note how the order of levels changes:

```
fac <- factor(c("Asia", "Asia", "West","West","West"))
levels(fac)
```

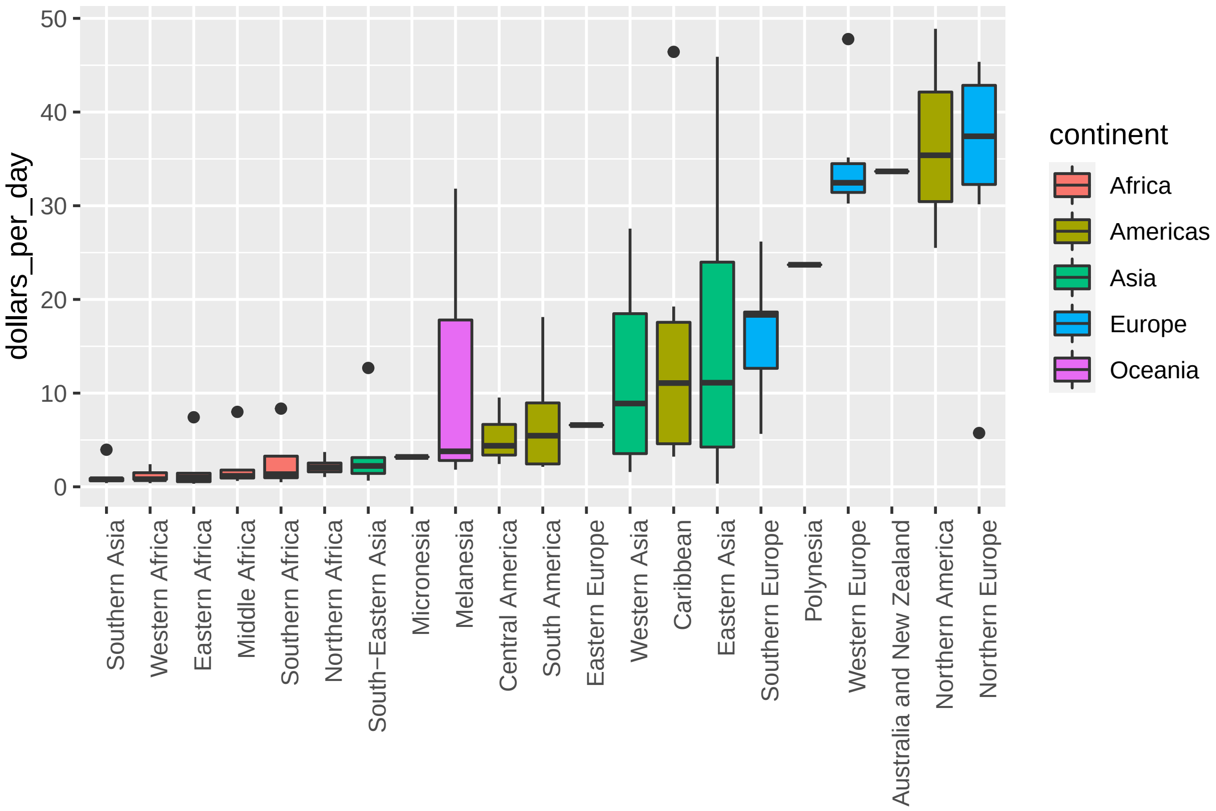
```
## [1] "Asia" "West"
```

```
value <- c(10,11,12,6,4)
fac <- reorder(fac,value,FUN = mean)
levels(fac)
```

```
## [1] "West" "Asia"
```

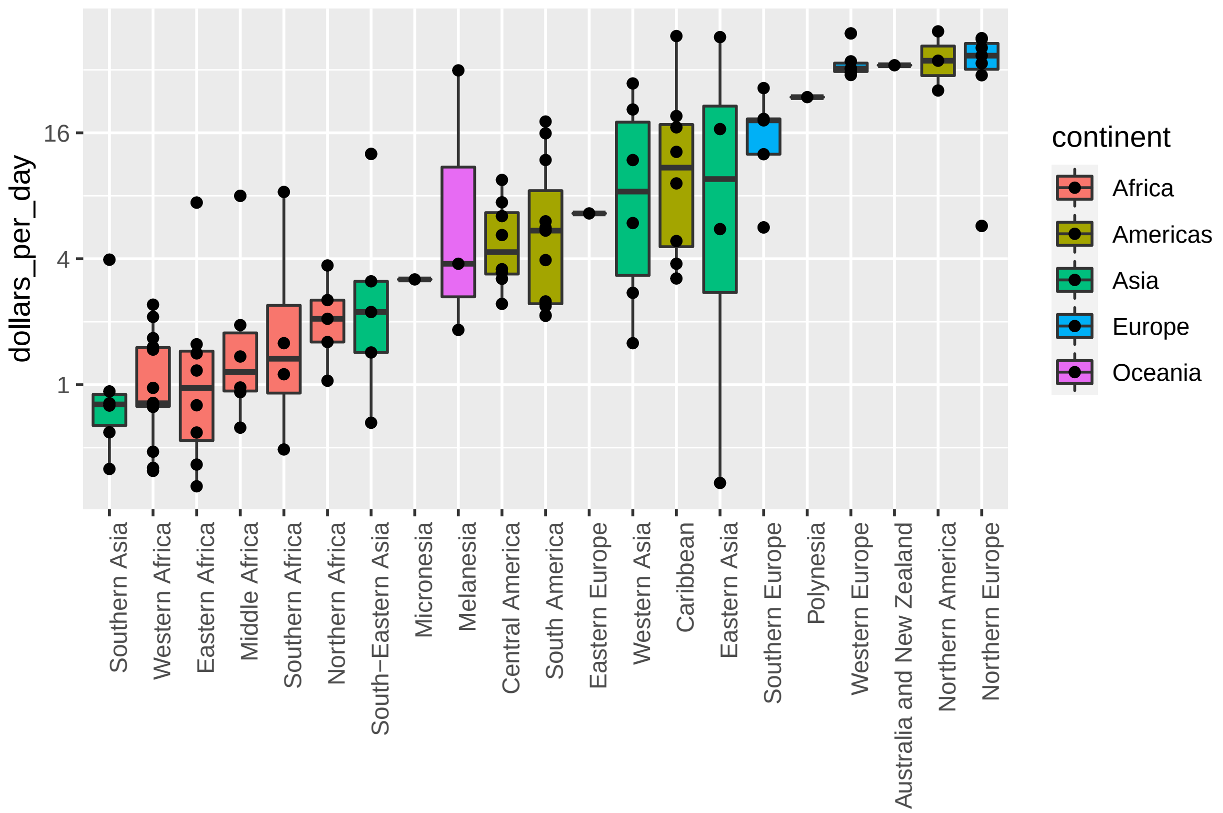
Second, we can use color to distinguish the different continents, a visual cue that helps find specific regions. Here is the code:

```
p <- gapminder %>%
  filter(year == past_year & !is.na(gdp)) %>%
  mutate(region = reorder(region, dollars_per_day, FUN = median)) %>%
  ggplot(aes(region, dollars_per_day, fill = continent)) +
  geom_boxplot() +
  theme(axis.text.x = element_text(angle = 90, hjust = 1)) +
  xlab("")
p
```



This plot shows two clear groups, with the rich group composed of North America, Northern and Western Europe, New Zealand and Australia. As with the histogram, if we remake the plot using a log scale

```
p + scale_y_continuous(trans = "log2") + geom_point()
```



we are able to better see differences within the devoloping world.

Show the data In many cases we do not show the data because it adds clutter to the plot and obfuscates the message. In the example above this is not the case and adding points actually lets us see all the data. We can add this layer using `geom_point()`.

Comparing Distributions

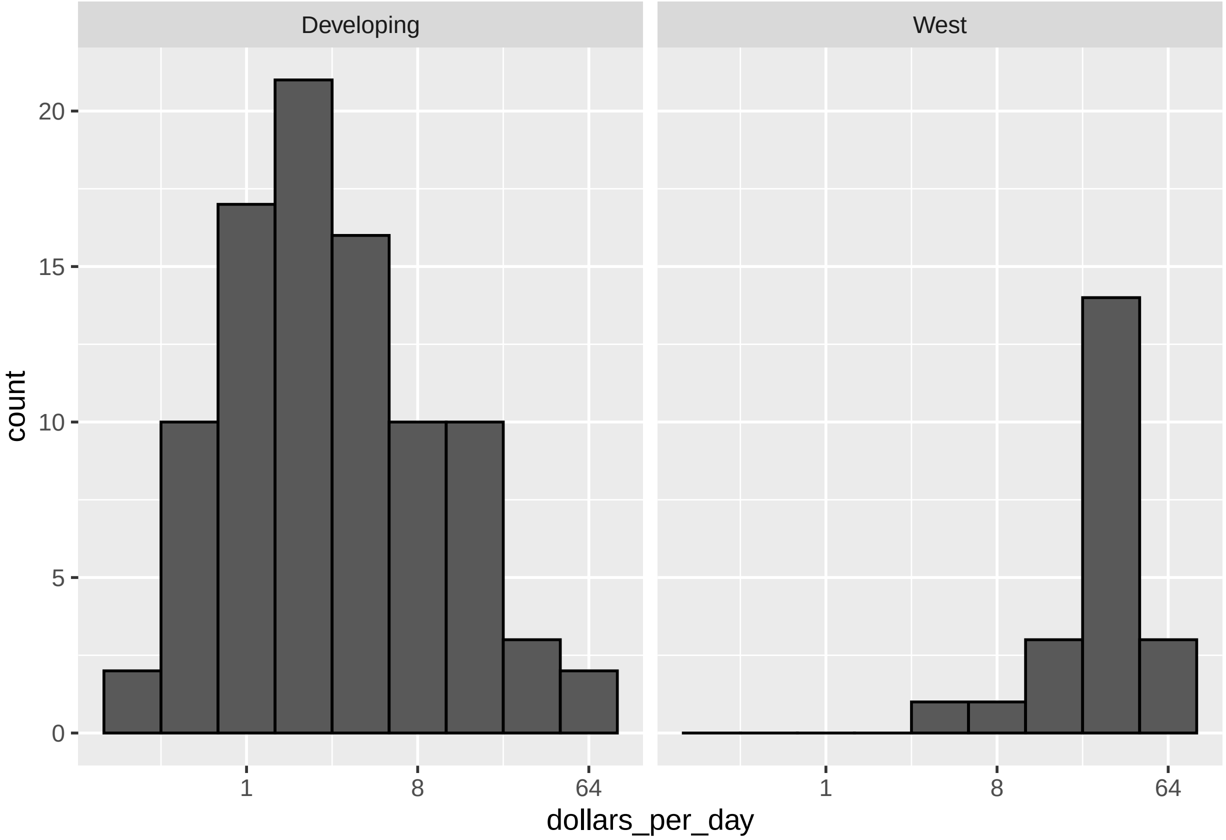
The exploratory data analysis above has revealed two characteristics about average income distribution in 1970. Using a histogram we found a bimodal distribution with the modes relating to poor and rich countries. Then by stratifying by region and examining boxplots we found that the rich countries were mostly in Europe and Northern America, along with Australia and New Zealand. We define a vector with these regions:

```
west <- c("Western Europe","Northern Europe","Southern Europe",
          "Northern America","Australia and New Zealand")
```

Now we want to focus on comparing the differences in distributions across time.

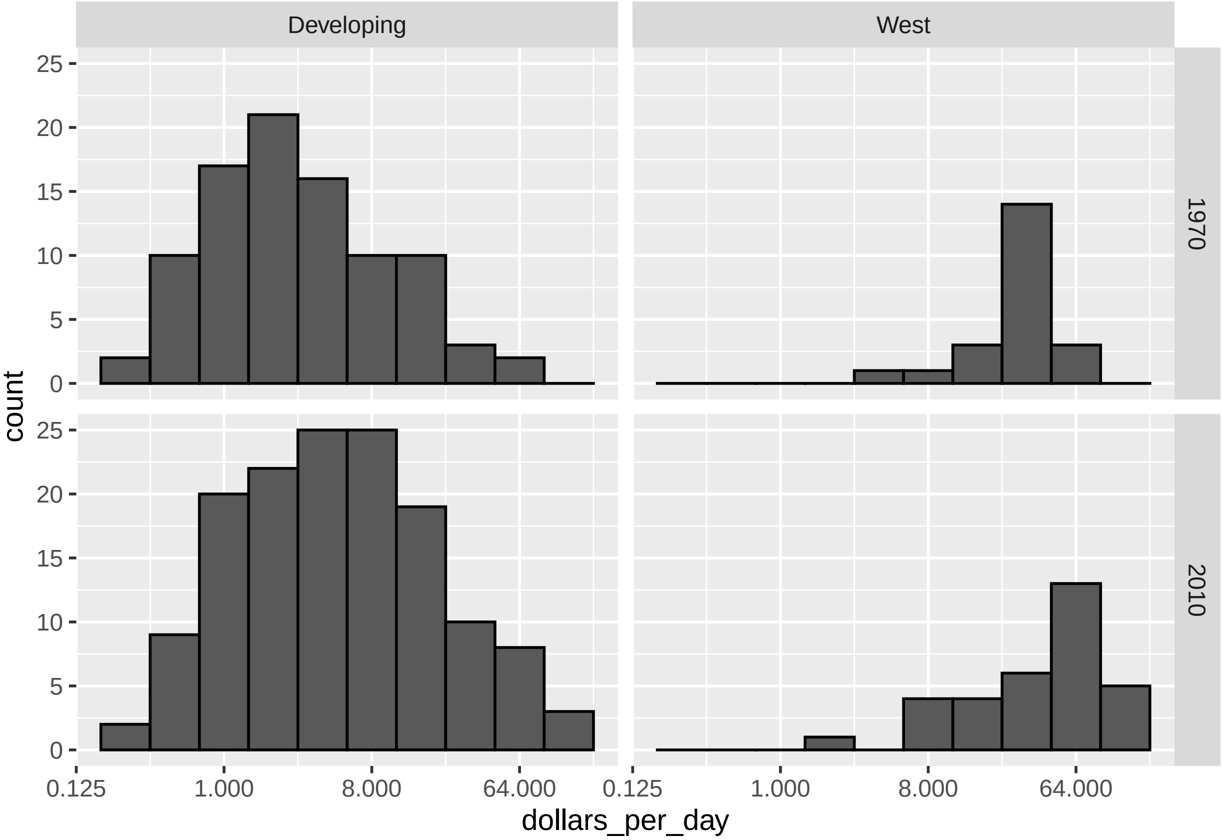
We start by confirming that the bimodality observed in 1970 is explained by a west versus devoloping world dichotomy. We do this by creating histograms for the groups previously identified. Note that we create the two groups with an `ifelse` inside a `mutate` and that we use `facet_grid` to make a histogram for each group:

```
gapminder %>%
  filter(year == past_year & !is.na(gdp)) %>%
  mutate(group = ifelse(region %in% west, "West", "Developing")) %>%
  ggplot(aes(dollars_per_day)) +
  geom_histogram(binwidth = 1, color = "black") +
  scale_x_continuous(trans = "log2") +
  facet_grid(. ~ group)
```



Now we are ready to see if the separation is worse today than it was 40 years ago. We do this by faceting by both region and year:

```
past_year <- 1970
present_year <- 2010
gapminder %>%
  filter(year %in% c(past_year, present_year) & !is.na(gdp)) %>%
  mutate(group = ifelse(region %in% west, "West", "Developing")) %>%
  ggplot(aes(dollars_per_day)) +
  geom_histogram(binwidth = 1, color = "black") +
  scale_x_continuous(trans = "log2") +
  facet_grid(year ~ group)
```



Before we interpret the findings of this plot, we note that there are more countries represented in the 2010 histograms than in 1970: the total counts are larger. One reason for this is that several countries were founded after 1970. For example, the Soviet Union turned into several countries including Russia and Ukraine during the 1990s. Another reason is that data was available for more countries during 2010.

We remake the plots using only countries with data available for both years. In the data wrangling chapter we will learn `tidyverse` tools that permit us to write efficient code for this, but here is simple code using the `intersect` function:

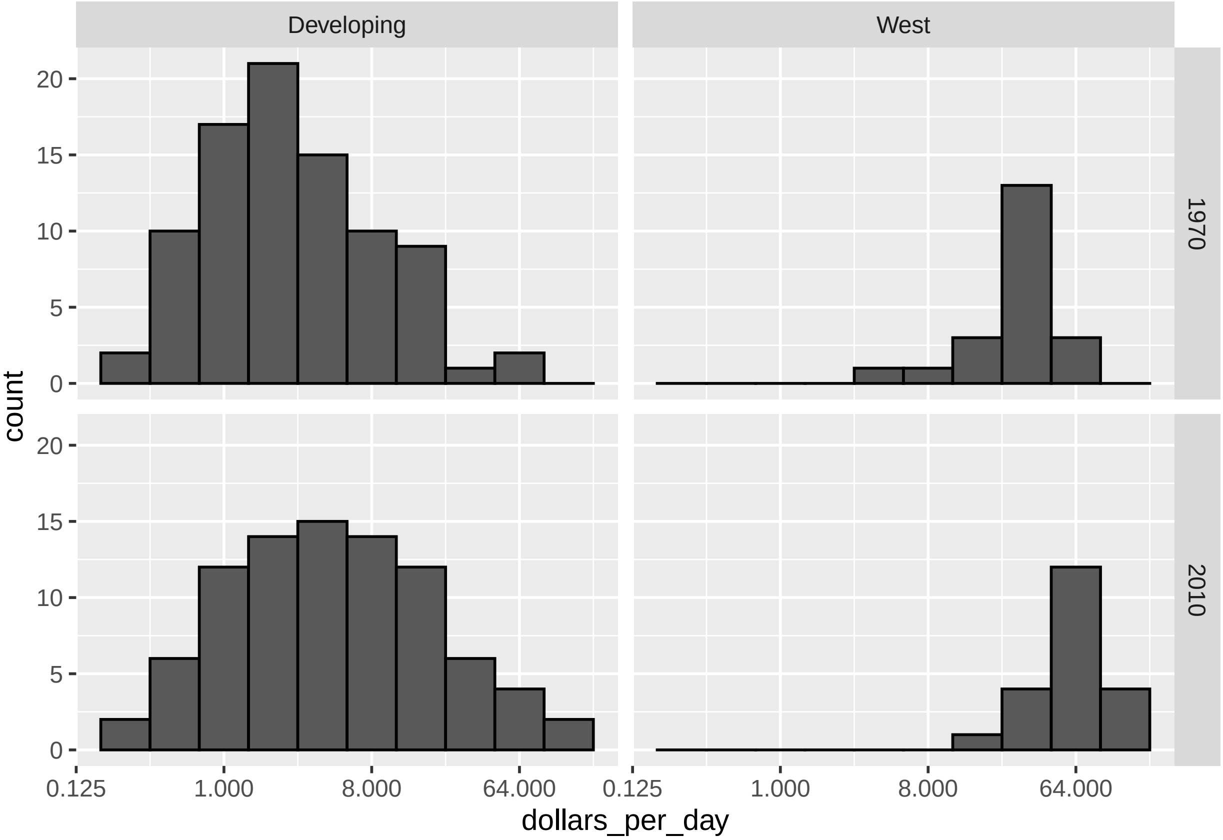
```
country_list_1 <- gapminder %>%
  filter(year == past_year & !is.na(dollars_per_day)) %>% .$country

country_list_2 <- gapminder %>%
  filter(year == present_year & !is.na(dollars_per_day)) %>% .$country

country_list <- intersect(country_list_1, country_list_2)
```

These 108 account for 86 % of the world population, so this subset should be representative.

Let's remake the plot but only for this subset by simply adding `country %in% country_list` to the filter



function:

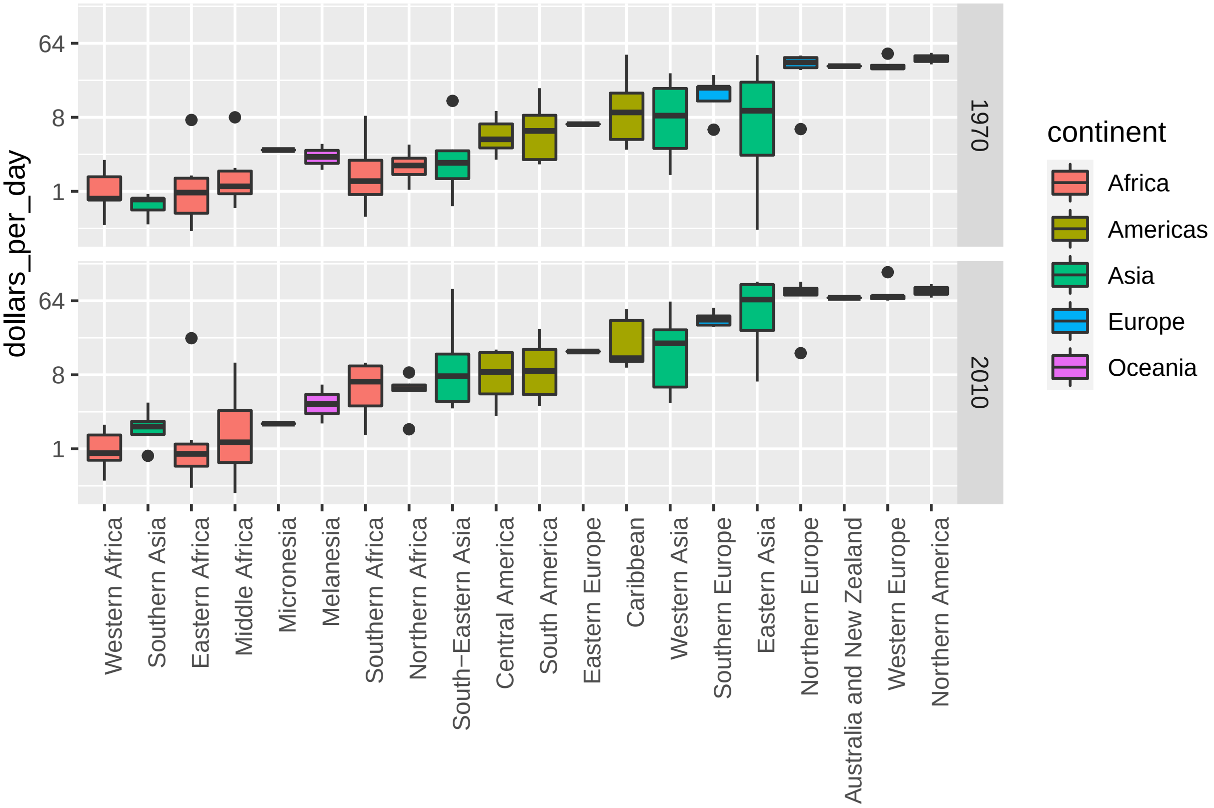
We now see that while the rich countries have become a bit richer, percentage-wise, the poor countries appear to have improved more. In particular we see that the proportion of *developing* countries earning more than $16 a day increases substantially.

To see which specific regions improved the most we can remake the boxplots we made above but now adding 2010

```
p <- gapminder %>%
  filter(year %in% c(past_year, present_year) & country %in% country_list) %>%
  mutate(region = reorder(region, dollars_per_day, FUN = median)) %>%
  ggplot() +
  theme(axis.text.x = element_text(angle = 90, hjust = 1)) +
  xlab("") +
  scale_y_continuous(trans = "log2")
```

and then using facet to compare the two years:

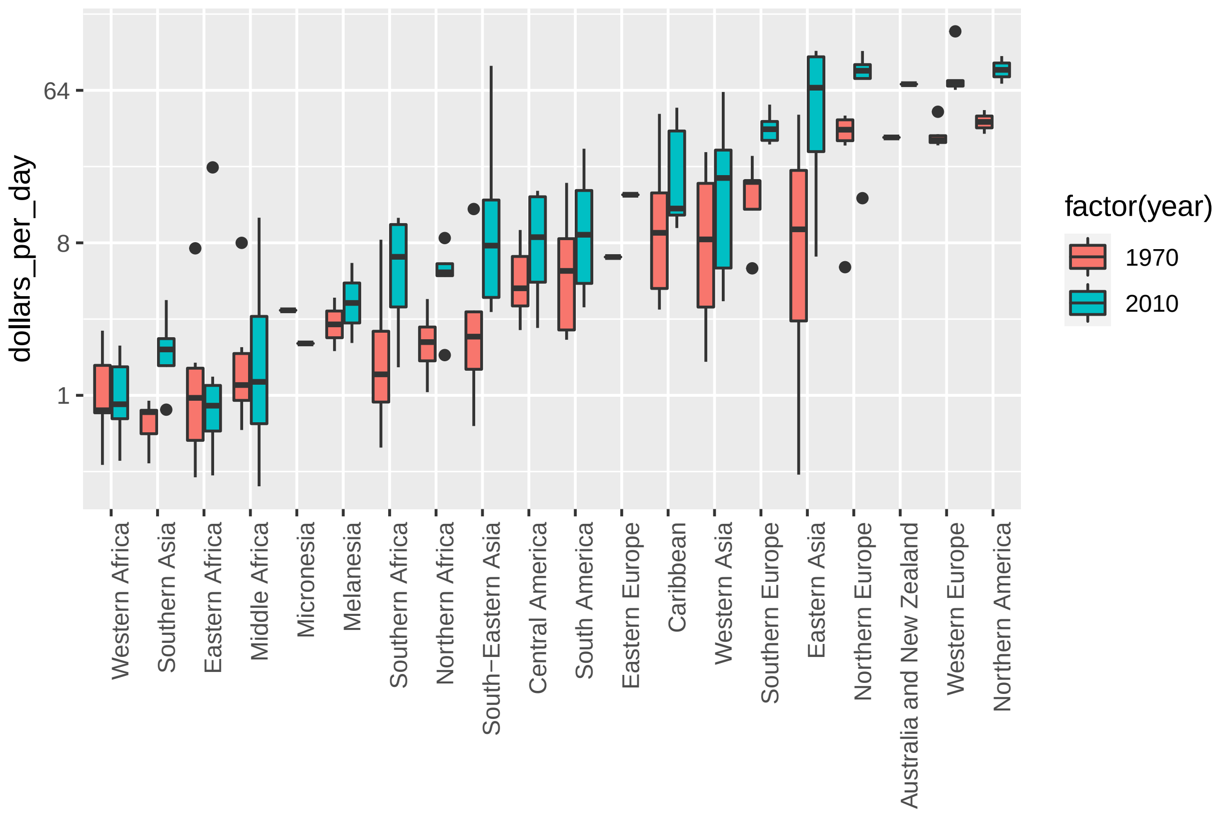
```
p + geom_boxplot(aes(region, dollars_per_day, fill = continent)) + facet_grid(year~.)
```



Here, we pause to introduce another powerful ggplot2 feature. Because we want to compare each region before and after, it would be convenient to have the 1970 boxplot next to the 2010 boxplot for each region. In general, comparisons are easier when data are plotted next to each other.

So, instead of faceting, we keep the data from each year together, but ask ggplot to color (or fill) them depending on the year. ggplot automatically separates them and puts the two boxplots next to each other. Because **year is a number, we turn it into a factor** because **ggplot automatically assigns a color to each category of a factor:**

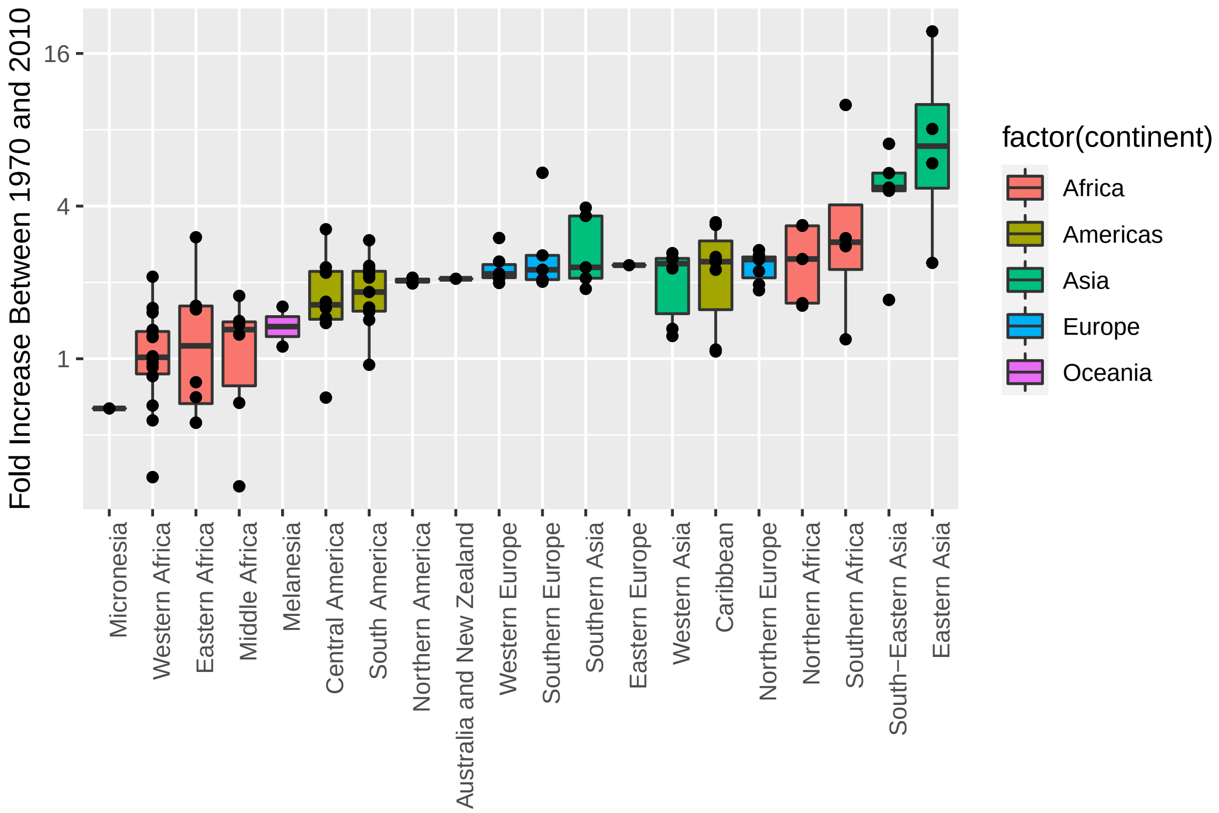
```
p + geom_boxplot(aes(region, dollars_per_day, fill = factor(year)))
```



Finally, we point out that if what we are most interested is in comparing before and after values, it might make more sense to plot the ratios, or difference in the log scale. We are still not ready to learn to code this but here is what the plot would look like:

```
##      country                    region continent year dollars_per_day
## 1    Algeria           Northern Africa    Africa past       3.7172265
## 2  Argentina             South America  Americas past      18.1207496
## 3  Australia Australia and New Zealand   Oceania past      33.6671656
## 4    Austria            Western Europe    Europe past      30.2348264
## 5    Bahamas                 Caribbean  Americas past      46.4254181
## 6 Bangladesh             Southern Asia      Asia past       0.7950843
```

```
##      country                    region continent       past   present
## 1    Algeria           Northern Africa    Africa  3.7172265  6.018638
## 2  Argentina             South America  Americas 18.1207496 28.871158
## 3  Australia Australia and New Zealand   Oceania 33.6671656 69.602975
## 4    Austria            Western Europe    Europe 30.2348264 73.114157
## 5    Bahamas                 Caribbean  Americas 46.4254181 50.493566
## 6 Bangladesh             Southern Asia      Asia  0.7950843  1.499445
```



To make the y-axis label general we could use the `paste` function:

```
ylab(paste("Fold Increase Between", past_year,"and",present_year))
```

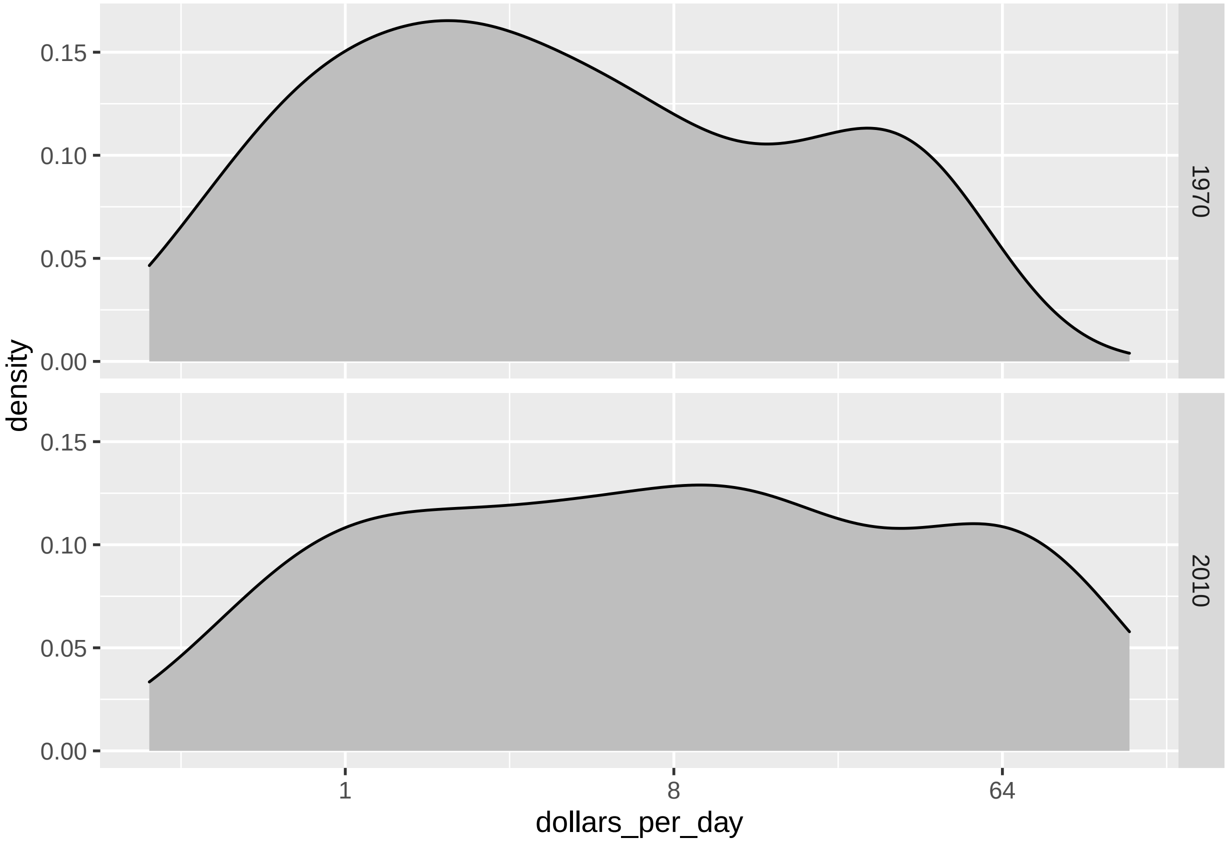
```
## $y
## [1] "Fold Increase Between 1970 and 2010"
##
## attr(,"class")
## [1] "labels"
```

Smooth Density Plots

We have used data exploration to discover that income gap between rich and poor countries has closed considerably during the last 40 years. We used a series of histograms and boxplots to see this. Here we suggest a succinct way to convey this message with just one plot. We will use smooth density plots.

Let's start by noting that density plots for income distribution in 1970 and 2010 deliver the message that the gap is closing:

```
gapminder %>%
  filter(year %in% c(past_year, present_year) & country %in% country_list) %>%
  ggplot(aes(dollars_per_day)) +
  geom_density(fill = "grey") +
  scale_x_continuous(trans = "log2") +
  facet_grid(year~.)
```



In the 1970 plot we see two clear modes, poor and rich countries. In 2010 it appears that some of the poor countries have shifted towards the right, closing the gap.

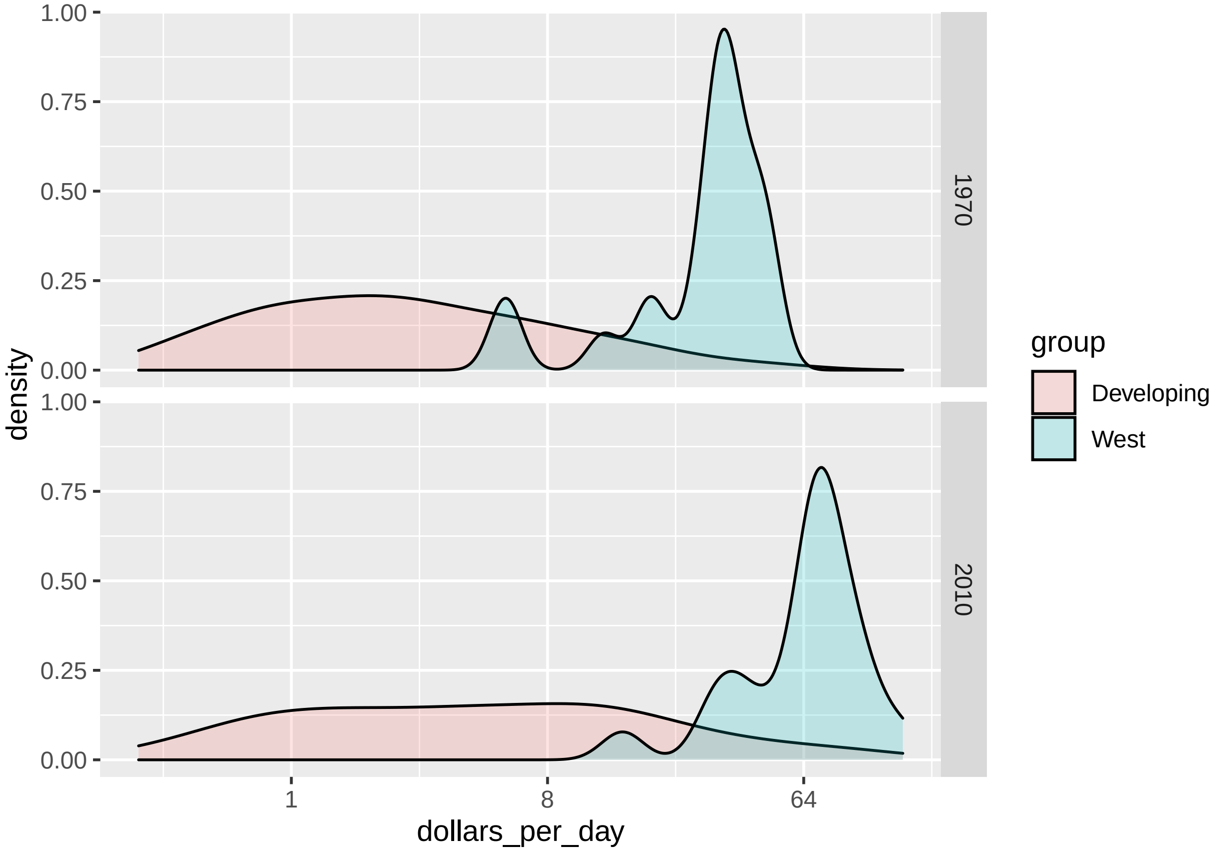
The next message we need to covey is that the reason for this change in distribution is that poor countries became richer rather than some rich countries becoming poorer. To do this all we need to do is assign a color to the groups we identified during data exploration.

However, before we can do this we need to learn how to make these smooth densities in a way that preserves information of how many countries are in each group. To understand why we need this, note the discrepancy in the size of each group:

group	n
Developing	87
West	21

but when we overlay two densities, the default is to have the area represented by each distribution add up to 1 regardless of the size of each group:

```
gapminder %>%
  filter(year %in% c(past_year, present_year) & country %in% country_list) %>%
  mutate(group = ifelse(region %in% west, "West", "Developing")) %>%
  ggplot(aes(dollars_per_day, fill = group)) +
  scale_x_continuous(trans = "log2") +
  geom_density(alpha = 0.2) +
  facet_grid(year ~ .)
```



which make it appear as if there are the same number of countries in each group. To change this, we will need to learn to access computed variables with the `geom_density` function.

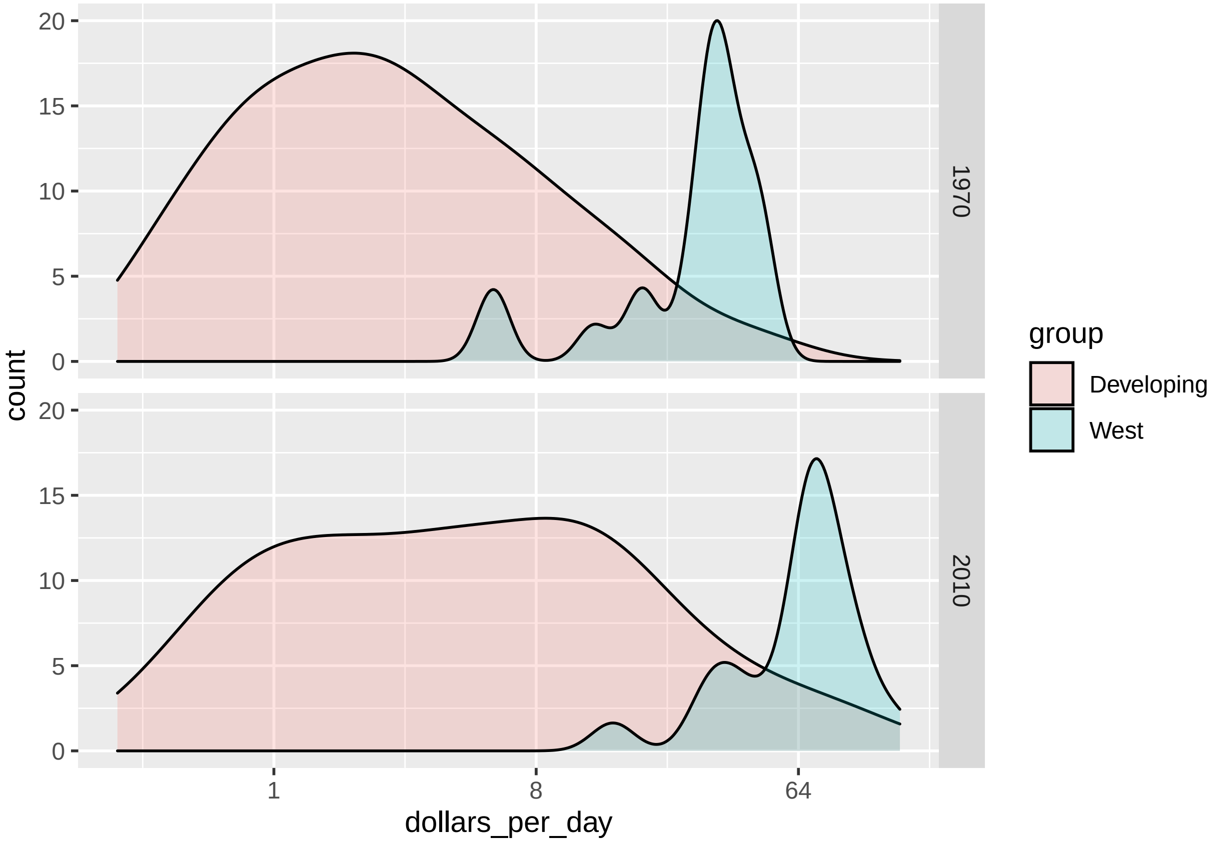
Accessing Computed Variables

To have the areas of these densities be proportional to the size of the groups, we can simply multiply the y-axis values by the size of the group. From the `geom_density` help file we see that the function computes a variable called `count` that does exactly this. We want this variable to be on the y-axis rather than the density.

In ggplot we access these variables by surrounding `count` by `..` (two periods). So we will use the following mapping:

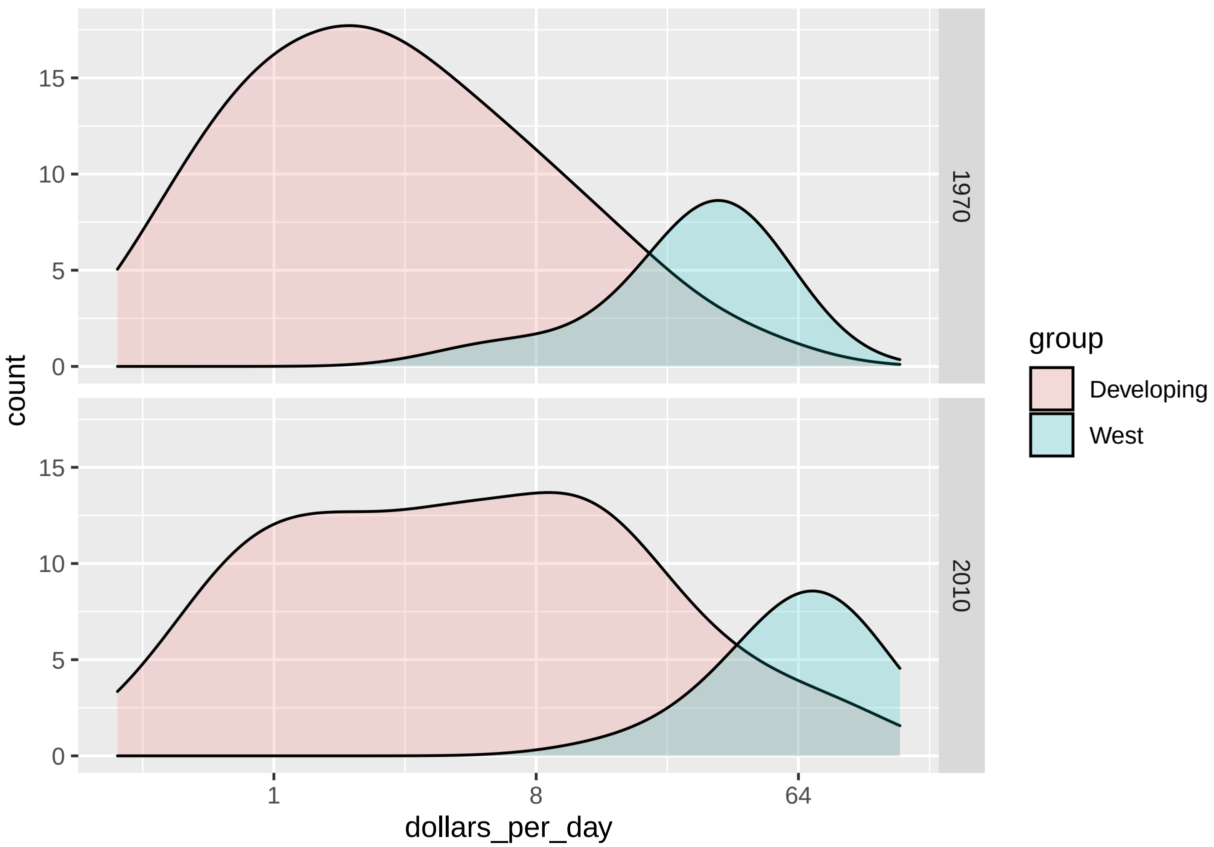
We can now create the desired plot by simply changing the mapping in the previous code chunk:

```
p <- gapminder %>%
  filter(year %in% c(past_year, present_year) & country %in% country_list) %>%
  mutate(group = ifelse(region %in% west, "West", "Developing")) %>%
  ggplot(aes(dollars_per_day, y = ..count.., fill = group)) +
  scale_x_continuous(trans = "log2")
p + geom_density(alpha = 0.2) + facet_grid(year ~ .)
```



If we want the densities to be smoother, we use the bw argument. We tried a few and decided on 0.75:

```
p + geom_density(alpha = 0.2, bw = 0.75) + facet_grid(year ~ .)
```



This plot now shows what is happening very clearly. The developing world distribution is changing. A third mode appears consisting of the countries that most closed the gap.

'case_when'

We can actually make this figure somewhat more informative. From the exploratory data analysis we noticed that many of the countries that most improved were from Asia. We can easily alter the plot to show key regions separately.

We introduce the `case_when` function useful for defining groups:

```
gapminder <- gapminder %>%
  mutate(group = case_when(    # can be like an ifelse() function for multiple conditions
    region %in% west ~ "West",
    region %in% c("Eastern Asia", "South-Eastern Asia") ~ "East Asia",
    region %in% c("Caribbean", "Central America", "South America") ~ "Latin America",
    continent == "Africa" & region != "Northern Africa" ~ "Sub-Saharan Africa",
    TRUE ~ "Others"))   # for all other regions that don't meet conditions above
```

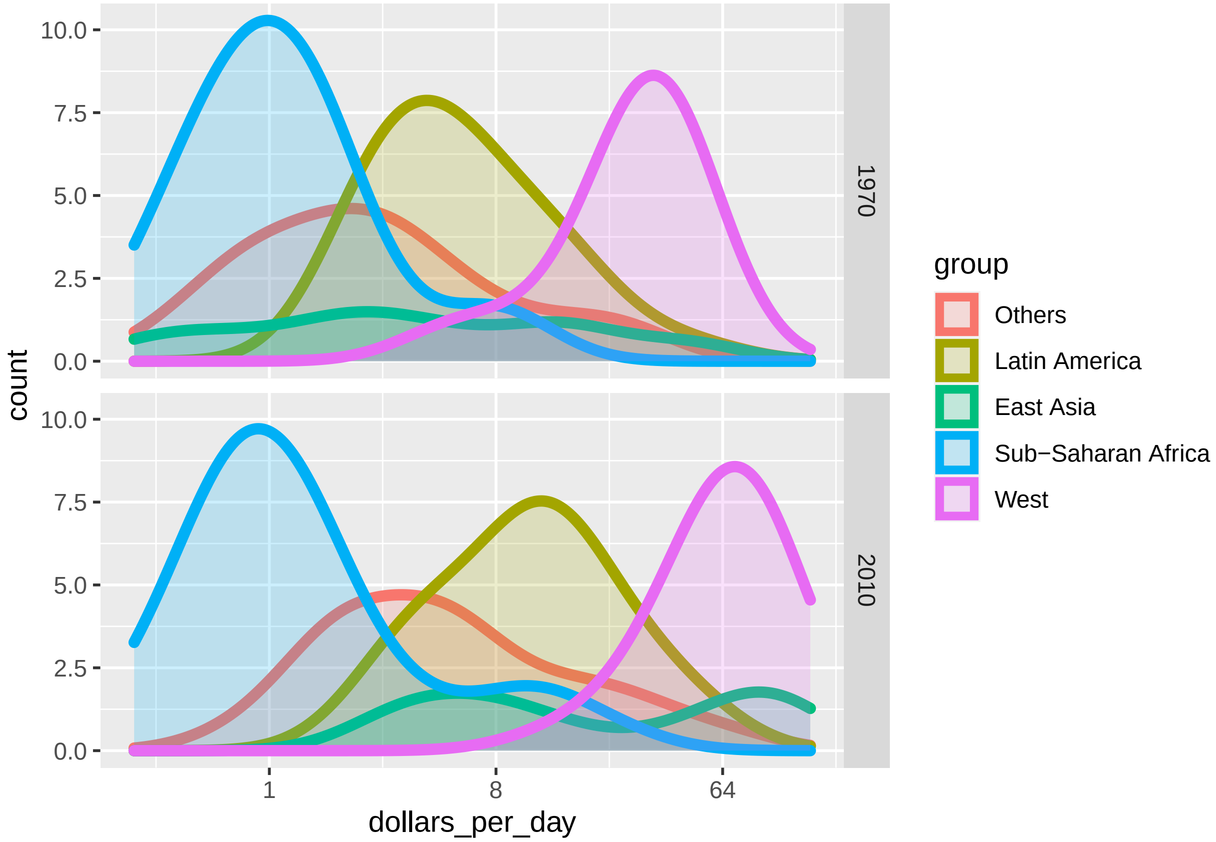
We turn this `group` variable into a factor to control the order of the levels:

```
gapminder <- gapminder %>%
  mutate(group = factor(group, levels = c("Others", "Latin America", "East Asia","Sub-Saharan Africa",
```

We pick this particular order for a reason that becomes clear later.

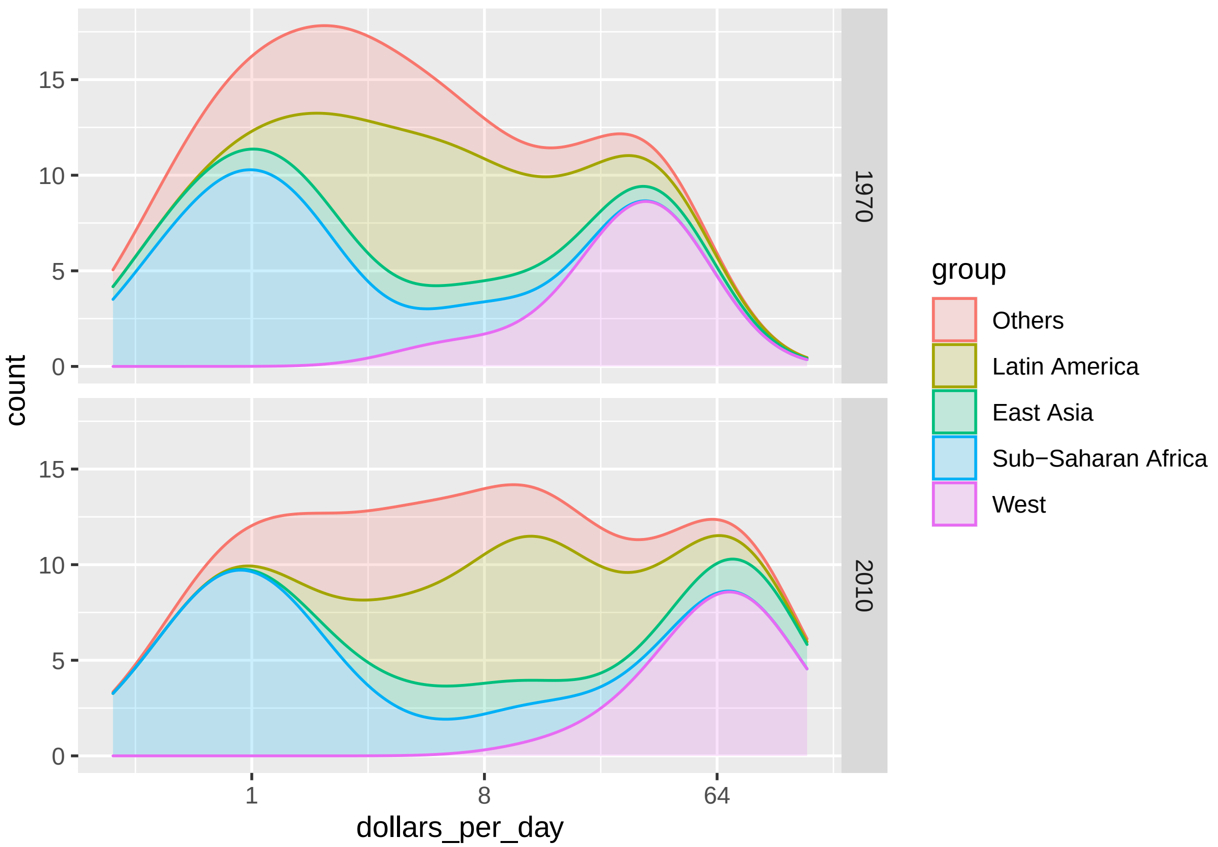
We can now easily plot the densities for each. We use `color` and `size` to clearly see the tops:

```
p <- gapminder %>%
    filter(year %in% c(past_year, present_year) & country %in% country_list) %>%
  ggplot(aes(dollars_per_day, y = ..count.., fill = group, color = group)) +
  scale_x_continuous(trans = "log2")
p + geom_density(alpha = 0.2, bw = 0.75, size = 2) + facet_grid(year ~ .)
```



The plot is cluttered and somewhat hard to read. A clearer picture is sometimes achieved by stacking the densities on top of each other:

```
p + geom_density(alpha = 0.2, bw = 0.75, position = "stack") + facet_grid(year ~.)
```

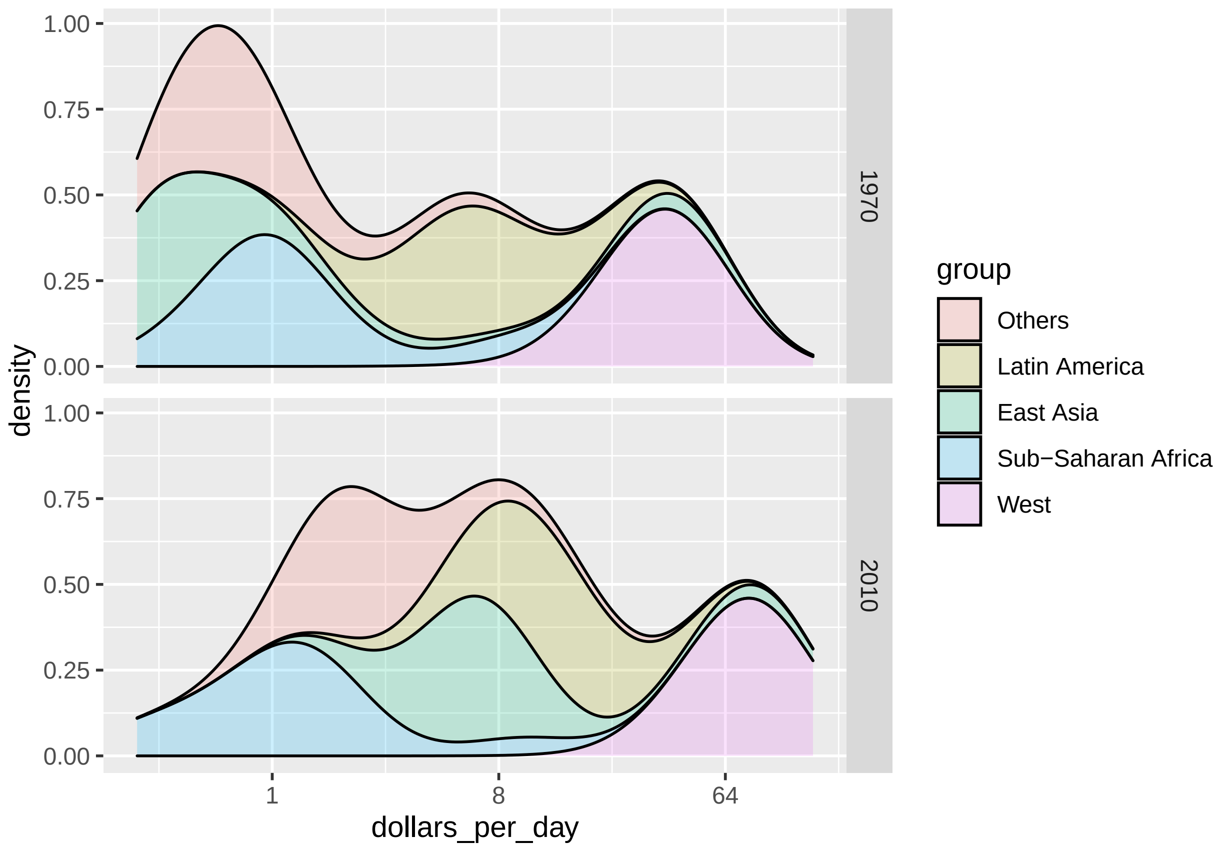


Here we can clearly see how the distributions for East Asia, Latin America and Others shift markedly to the right. While Sub-Saharan Africa remains stagnant.

Note that we order the levels of the group so that The West density be plotted first, then Sub-Saharan Africa. *Having the two extremes be plotted first let's us see the remaining bimodality better.*

Weighted densities As a final point, we note that these distributions weigh every country the same. So if most of the population is improving, but living in a very large country, such as China, we might not appreciate this. We can actually weight the smooth densities using the `weight` mapping argument. Here we weight by population:

```
gapminder %>%
    filter(year %in% c(past_year, present_year) & country %in% country_list) %>%
  group_by(year) %>%
  mutate(weight = population/sum(population)) %>%
  ungroup() %>%
  ggplot(aes(dollars_per_day, fill = group, weight = weight)) +
  scale_x_continuous(trans = "log2") +
  geom_density(alpha = 0.2, bw = 0.75, position = "stack") + facet_grid(year ~ .)
```



This particular figure shows very clearly how the income distribution gap is closing with most of the poor remaining in Sub-Saharan Africa.

Ecological Fallacy

Throughout this section we have been comparing regions of the world. We have seen that on average, some regions do better than others. Now we focus on describing the importance of variability within the groups.

Here we will focus on the relationship between country child survival rates and average income. We start by comparing these quantities across regions. We define a few more regions:

```
gapminder <- gapminder %>%
  mutate(group = case_when(
    region %in% west ~ "The West",
    region %in% "Northern Africa" ~ "Northern Africa",
    region %in% c("Eastern Asia", "South-Eastern Asia") ~ "East Asia",
    region == "Southern Asia"~ "Southern Asia",
    region %in% c("Central America", "South America", "Caribbean") ~ "Latin America",
    continent == "Africa" & region != "Northern Africa" ~ "Sub-Saharan Africa",
    region %in% c("Melanesia", "Micronesia", "Polynesia") ~ "Pacific Islands"))
```

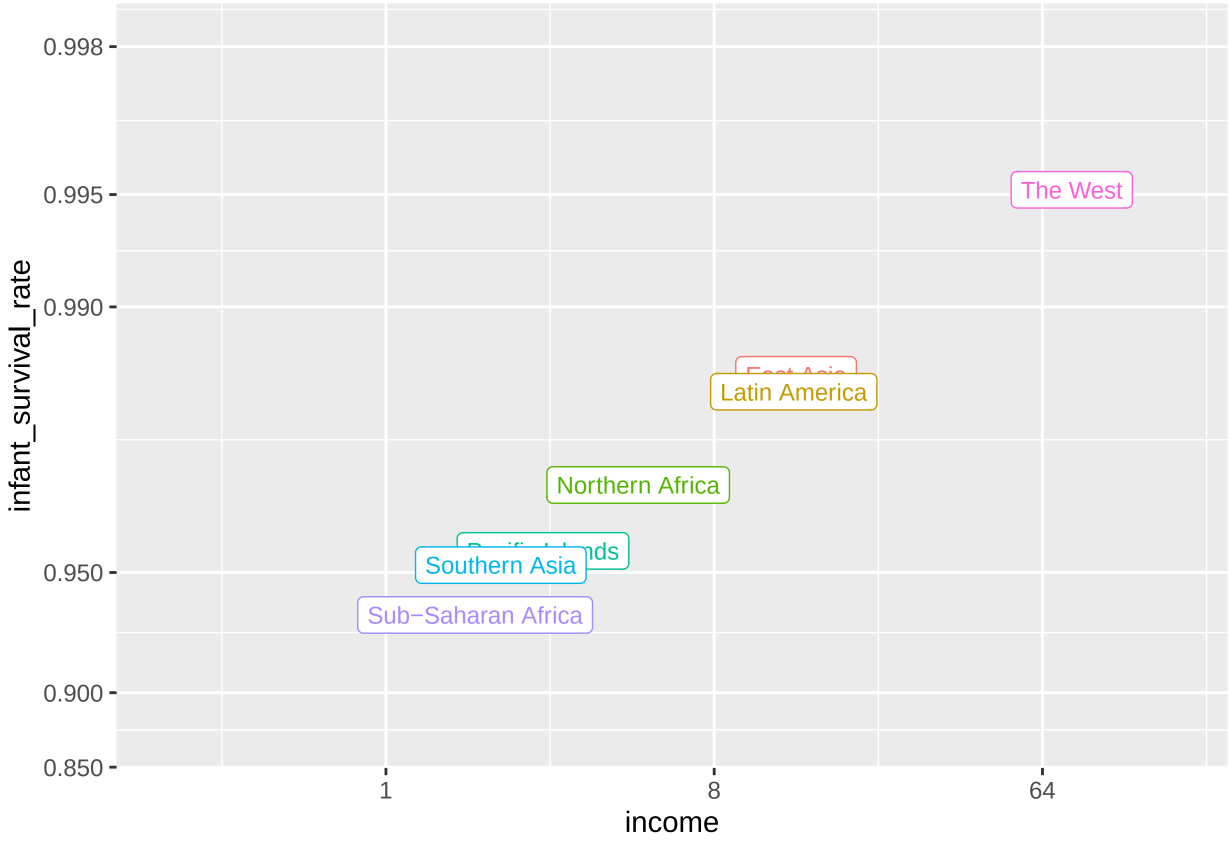
We then compute these quantities for each region.

```
surv_income <- gapminder %>%
  filter(year %in% present_year & !is.na(gdp) & !is.na(infant_mortality) & !is.na(group)) %>%
  group_by(group) %>%
  summarize(income = sum(gdp)/sum(population)/365,
            infant_survival_rate = 1-sum(infant_mortality/1000*population)/sum(population))
surv_income %>% arrange(income)
```

```
## # A tibble: 7 x 3
##   group              income infant_survival_rate
##   <chr>               <dbl>                <dbl>
## 1 Sub-Saharan Africa   1.76                0.936
## 2 Southern Asia        2.07                0.952
## 3 Pacific Islands      2.70                0.956
## 4 Northern Africa      4.94                0.970
## 5 Latin America       13.2                 0.983
## 6 East Asia           13.4                 0.985
## 7 The West            77.1                 0.995
```

This shows a dramatic difference. While in the west less than 0.5% children die, in Sub-Saharan Africa the rate is higher than 6%! The relationship between these two variables is almost perfectly linear.

```
surv_income %>% ggplot(aes(income, infant_survival_rate, label = group, color = group)) +
  scale_x_continuous(trans = "log2", limit = c(0.25, 150)) +
  scale_y_continuous(trans = "logit", limit = c(0.875, .9981),
                     breaks = c(.85,.90,.95,.99,.995,.998)) +
  geom_label(size = 3, show.legend = FALSE)
```



In this plot we introduce the use of the `limit` argument which lets us change the range of the axes. We are making the range larger than the data needs because we will later compare this plot to one with more variability and we want the ranges to be the same. We also introduce the `breaks` argument which lets us set the location of the axis tick labels. Finally we introduce a new transformation, the logistic transformation.

Logistic transformation The logistic or logit transformation for a proportion or rate p is defined as

$$f(p) = \log\left(\frac{p}{1-p}\right)$$

When p is a proportion or probability, the quantity that is being logged, $p/(1-p)$ is called the *odds*. In this case p is the proportion of children that survived. The odds tell us how many more children are expected to survive than to die. The log transformation makes this symmetric. If the rates are the same, then the log odds is 0. Fold increases or decreases turn into positive and negative increments respectively.

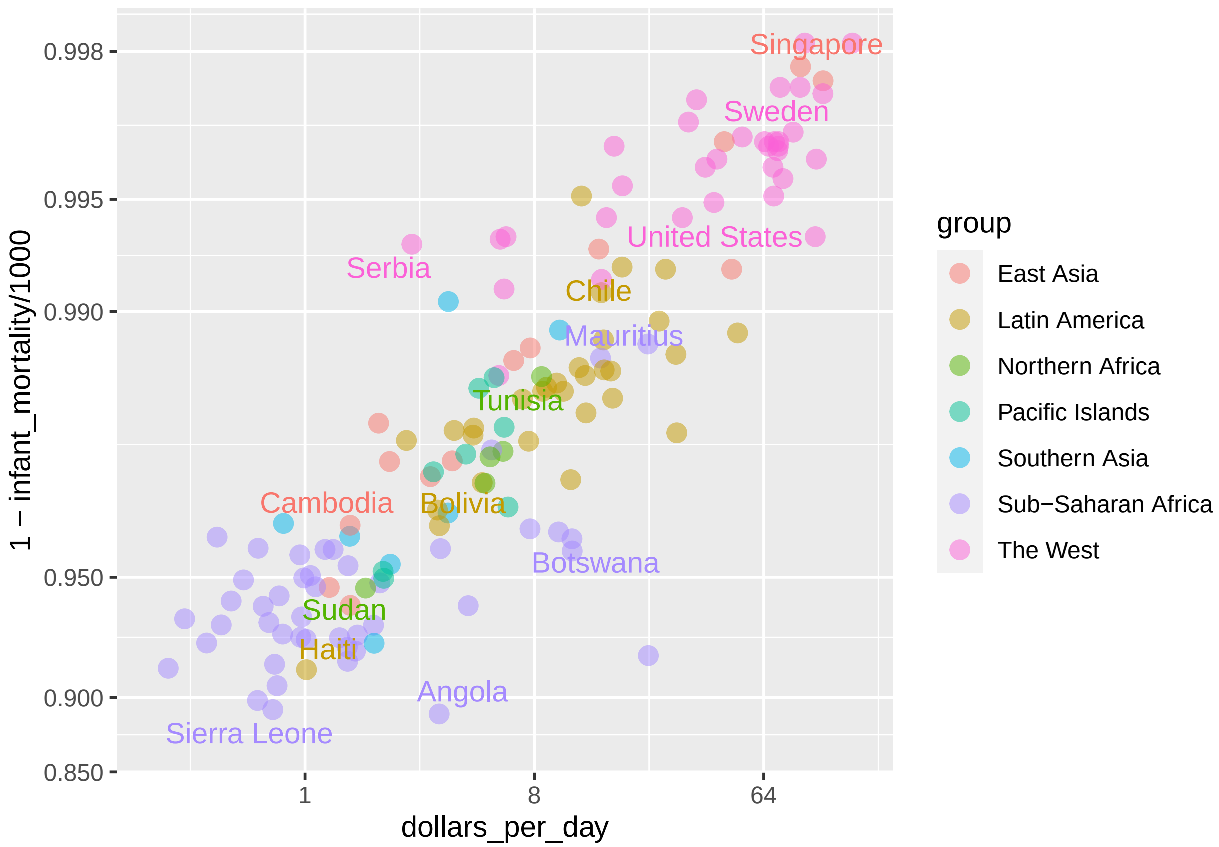
This scale is useful when we want to highlight differences near 0 or 1. For survival rates this is important because a survival rate of 90% is unacceptable, while a survival of 99% is relatively good. We would much prefer a survival rate closer to 99.9%. We want our scale to highlight these differences and the logit does this. Note that 99.9/0.1 is about 10 times bigger than 99/1 which is about 10 times larger than 90/10. And by using the log, these fold changes turn into constant increases.

Show the data

Now, back to our plot. Based on the plot above, do we conclude that a country with a low income is destined to have low survival rate? Do we conclude that all survival rates in Sub-Saharan Africa are all lower than in Southern Asia which in turn are lower than in the Pacific Islands, and so on?

Jumping to this conclusion based on a plot showing averages is referred to as the *ecological fallacy*. The almost perfect relationship between survival rates and income is only observed for the averages at the region level. Once we show all the data we see a somewhat more complicated story:

```r
library(ggrepel)
highlight <- c("Sierra Leone", "Mauritius",  "Sudan", "Botswana", "Tunisia",
               "Cambodia","Singapore","Chile", "Haiti", "Bolivia",
               "United States","Sweden", "Angola", "Serbia")
gapminder %>% filter(year %in% present_year & !is.na(gdp) & !is.na(infant_mortality) & !is.na(group) ) %
  ggplot(aes(dollars_per_day, 1 - infant_mortality/1000, color = group, label = country)) +
  scale_x_continuous(trans = "log2", limits=c(0.25, 150)) +
  scale_y_continuous(trans = "logit",limit=c(0.875, .9981),
                     breaks=c(.85,.90,.95,.99,.995,.998)) +
  geom_point(alpha = 0.5, size = 3) +
  geom_text_repel(size = 4, show.legend = FALSE,
    data = filter(gapminder, year %in% present_year & country %in% highlight))
```


Singapore
Sweden
United States
Serbia
Chile
Mauritius
Tunisia
Cambodia
Bolivia
Botswana
Sudan
Haiti
Angola
Sierra Leone
0.998
0.995
0.990
0.950
0.900
0.850
1 − infant_mortality/1000
1
8
64
dollars_per_day
group
East Asia
Latin America
Northern Africa
Pacific Islands
Southern Asia
Sub−Saharan Africa
The West