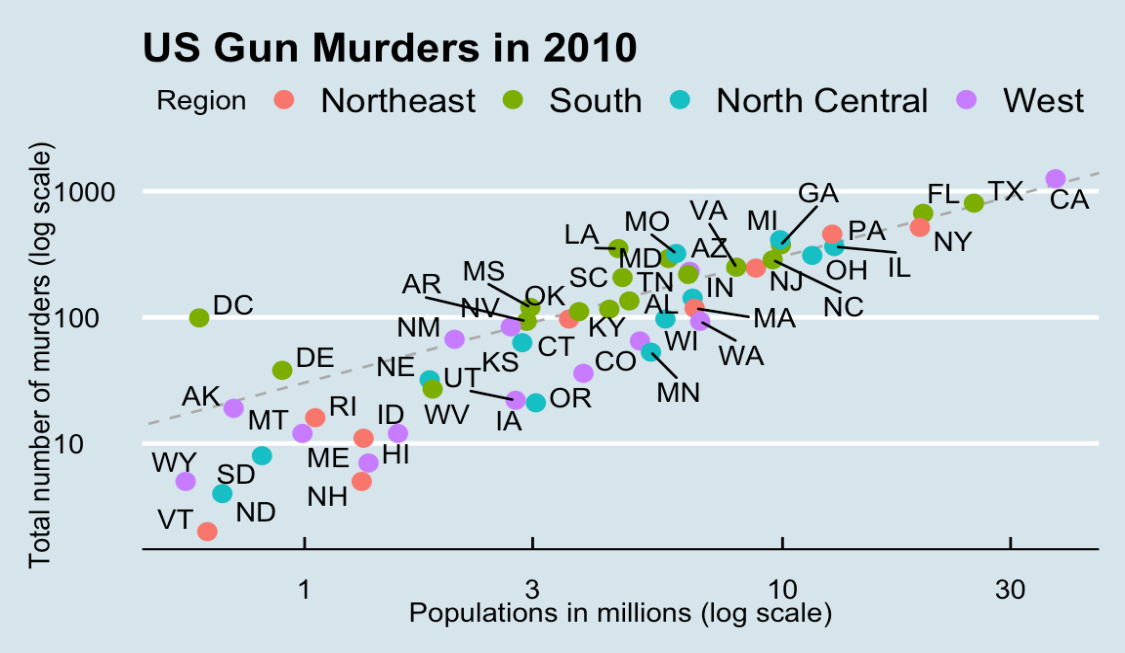
IV – Unit (Data Science)

I.Introduction to data visualization

Looking at the numbers and character strings that define a dataset is rarely useful. To convince yourself, print and stare at the US murders data table:

```
library(dslabs)
data(murders)
head(murders)
#>       state abb region population total
#> 1   Alabama  AL  South    4779736   135
#> 2    Alaska  AK   West     710231    19
#> 3   Arizona  AZ   West    6392017   232
#> 4  Arkansas  AR  South    2915918    93
#> 5 California  CA   West   37253956  1257
#> 6  Colorado  CO   West    5029196    65
```

What do you learn from staring at this table? How quickly can you determine which states have the largest populations? Which states have the smallest? How large is a typical state? Is there a relationship between population size and total murders? How do murder rates vary across regions of the country? For most human brains, it is quite difficult to extract this information just by looking at the numbers. In contrast, the answer to all the questions above are readily available from examining this plot:



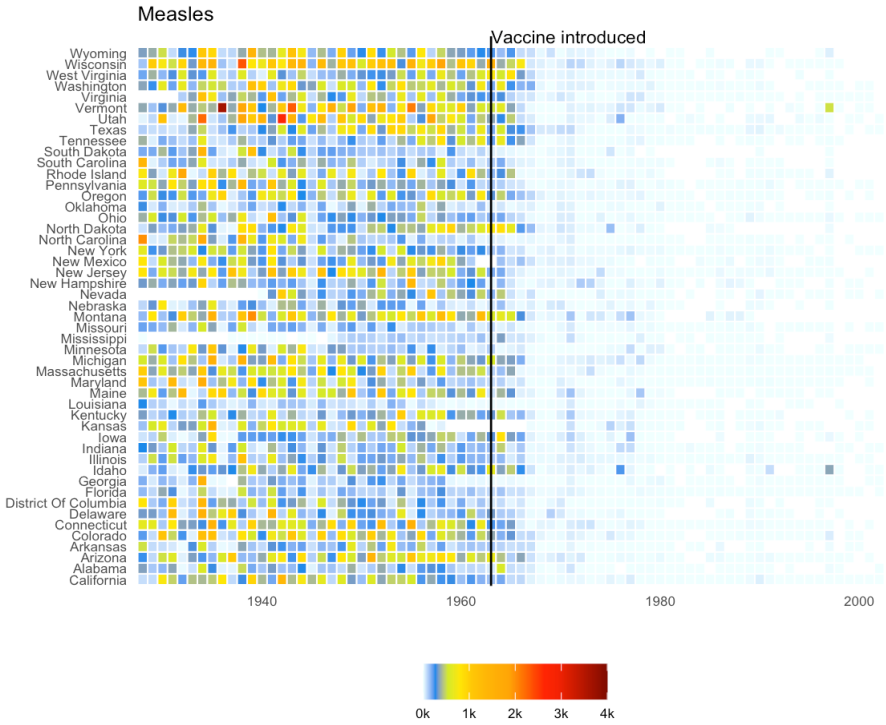
We are reminded of the saying “a picture is worth a thousand words”. Data visualization provides a powerful way to communicate a data-driven finding. In some cases, the visualization is so convincing that no follow-up analysis is required.

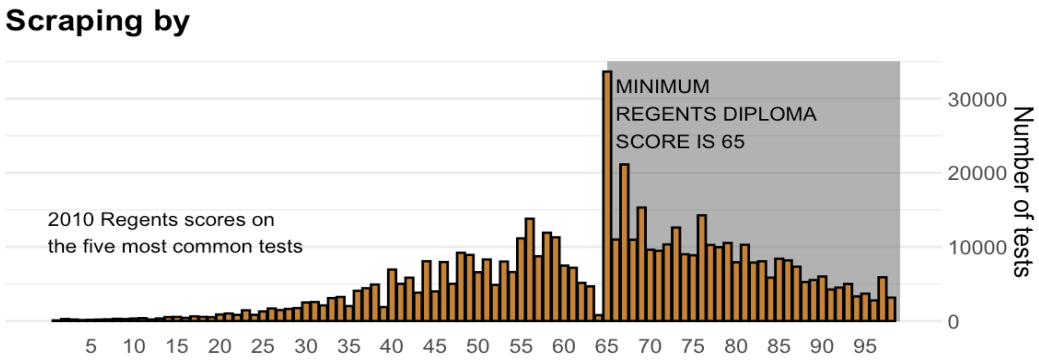
The growing availability of informative datasets and software tools has led to increased reliance on data visualizations across many industries, academia, and government. A salient example is news organizations, which are increasingly embracing *data journalism* and including effective *infographics* as part of their reporting.

A particularly effective example is a Wall Street Journal article[16] showing data related to the impact of vaccines on battling infectious diseases. One of the graphs shows measles cases by

US state through the years with a vertical line demonstrating when the vaccine was introduced.

```
#> Warning: Using `size` aesthetic for lines was
#> deprecated in ggplot2 3.4.0.
#> ℹ Please use `linewidth` instead.
#> This warning is displayed once every 8
#> hours.
#> Call
#> `lifecycle::last_lifecycle_warnings()`
#> to see where this warning was generated.
```





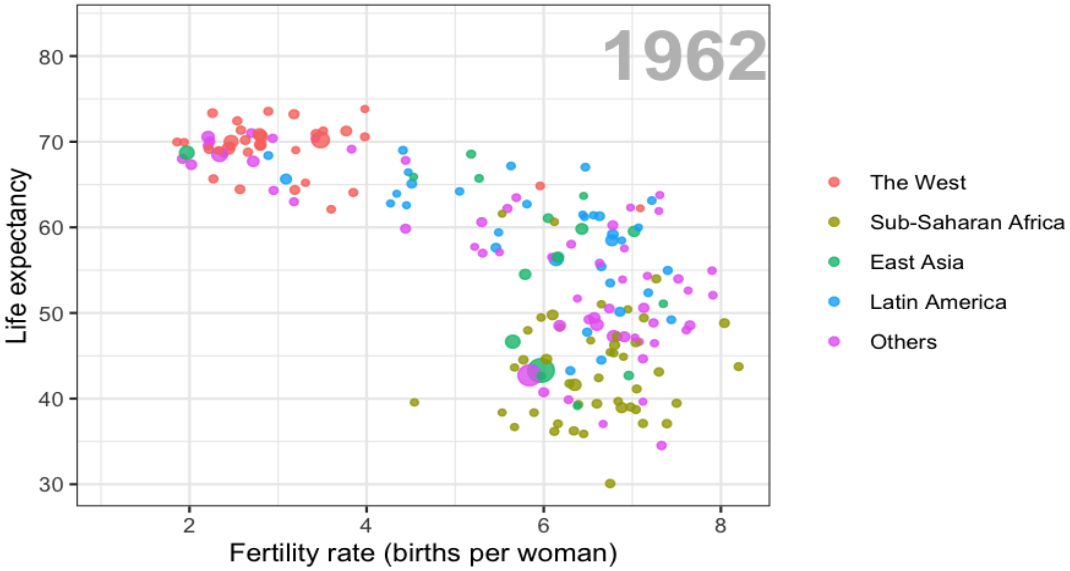
The most common test score is the minimum passing grade, with very few scores just below the threshold. This unexpected result is consistent with students close to passing having their scores bumped up.

This is an example of how data visualization can lead to discoveries which would otherwise be missed if we simply subjected the data to a battery of data analysis tools or procedures. Data visualization is the strongest tool of what we call *exploratory data analysis* (EDA). John W. Tukey[19], considered the father of EDA, once said,

"The greatest value of a picture is when it forces us to notice what we never expected to see."

Many widely used data analysis tools were initiated by discoveries made via EDA. EDA is perhaps the most important part of data analysis, yet it is one that is often overlooked.

Data visualization is also now pervasive in philanthropic and educational organizations. In the talks New Insights on Poverty[20] and The Best Stats You've Ever Seen[21], Hans Rosling forces us to notice the unexpected with a series of plots related to world health and economics. In his videos, he uses animated graphs to show us how the world is changing and how old narratives are no longer true.



II. Data visualization in practice

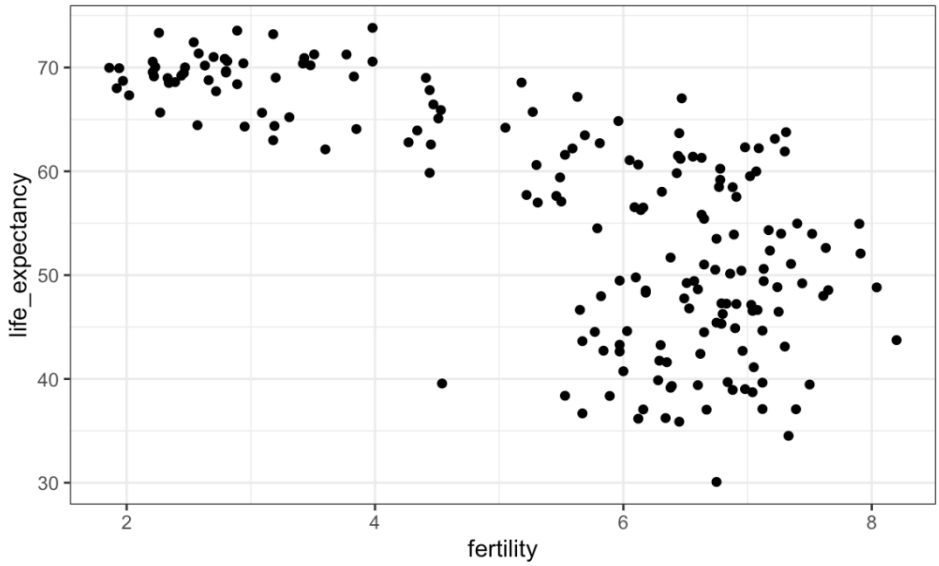
In this chapter, we will demonstrate how relatively simple **ggplot2** code can create insightful and aesthetically pleasing plots. As motivation we will create plots that help us better understand trends in world health and economics. We will implement what we learned in Chapters 8 and 9.8 and learn how to augment the code to perfect the plots. As we go through our case study, we will describe relevant general data visualization principles and learn concepts such as *faceting*, *time series plots*, *transformations*, and *ridge plots*.

1 Scatterplots

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 large families. But do the data support this dichotomous view?

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 tackle this challenge.
In order to analyze this world view, our first plot is a scatterplot of life expectancy versus fertility rates (average number of children per woman). We start by looking at data from about 50 years ago, when perhaps this view was first cemented in our minds.

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

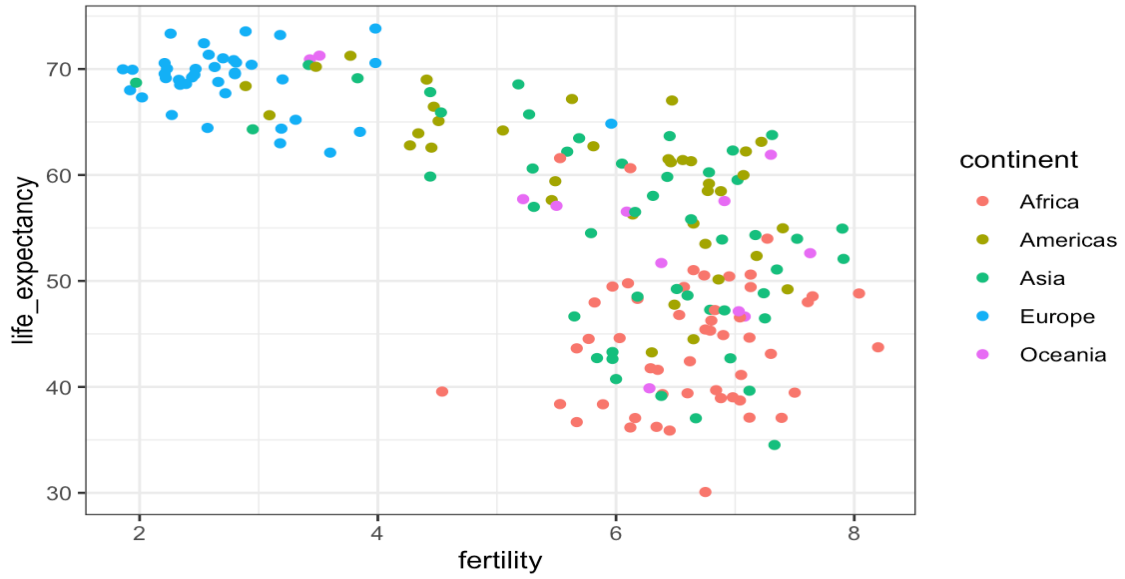


Most points fall into two distinct categories:

1. Life expectancy around 70 years and 3 or fewer children per family.
2. Life expectancy lower than 65 years and 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.

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



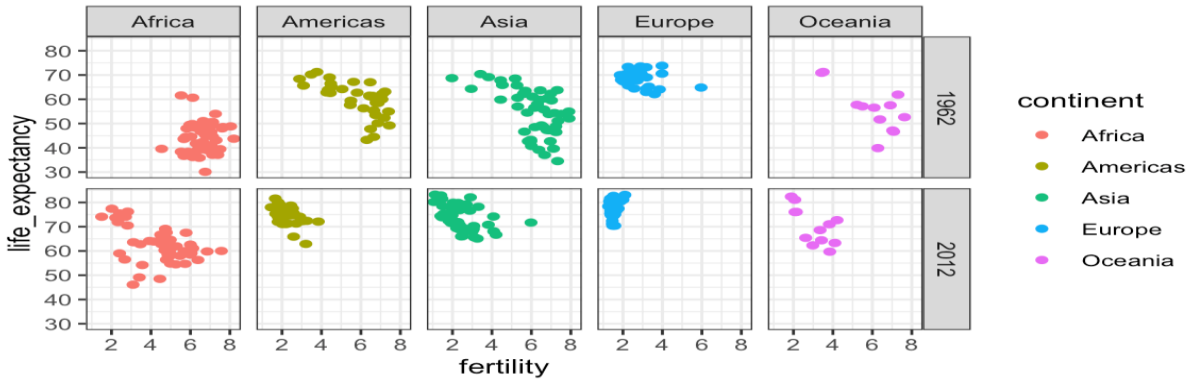
In 1962, “the West versus developing world” view was grounded in some reality. Is this still the case 50 years later?

2 Faceting

We could easily plot the 2012 data in the same way we did for 1962. To make comparisons, however, side by side plots are preferable.

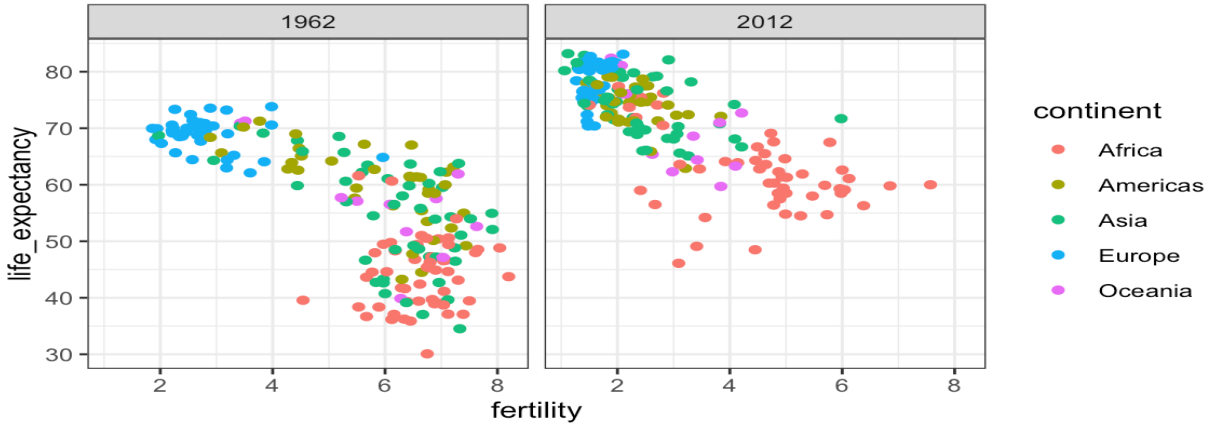
To achieve faceting, we add a layer with the function facet_grid, which automatically separates the plots. This function lets you facet by up to two variables using columns to represent one variable and rows to represent the other.

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



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:

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



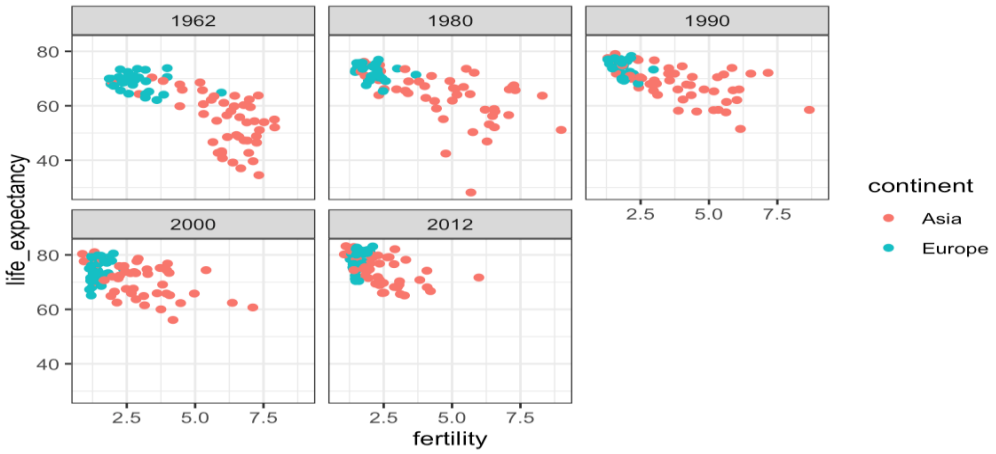
This plot clearly shows that the majority of countries have moved from the *developing world* cluster to the *western world* one. In 2012, the western versus developing world view no longer makes sense. This is particularly clear when comparing Europe to Asia, the latter of which includes several countries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

3 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, since 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 by automatically wrapping the series of plots so that each display has viewable dimensions:

```
years <- c(1962, 1980, 1990, 2000, 2012)
continents <- c("Europe", "Asia")
gapminder |>
  filter(year %in% years & continent %in% continents) |>
```

```
ggplot( aes(fertility, life_expectancy, col = continent)) +
geom_point() +
facet_wrap(~year)
```

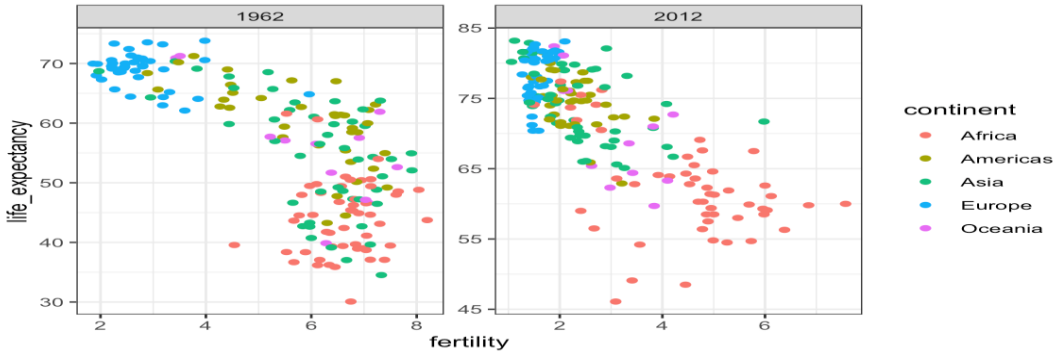


This plot clearly shows how most Asian countries have improved at a much faster rate than European ones.

4 Fixed scales for better comparisons

The default choice of the range of the axes is important. When not using `facet`, this range is determined by 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 above plot, we can 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 adjust the scales:

```
filter(gapminder, year%in%c(1962, 2012)) |>
 ggplot(aes(fertility, life_expectancy, col = continent)) +
 geom_point() +
 facet_wrap(. ~ year, scales = "free")
```



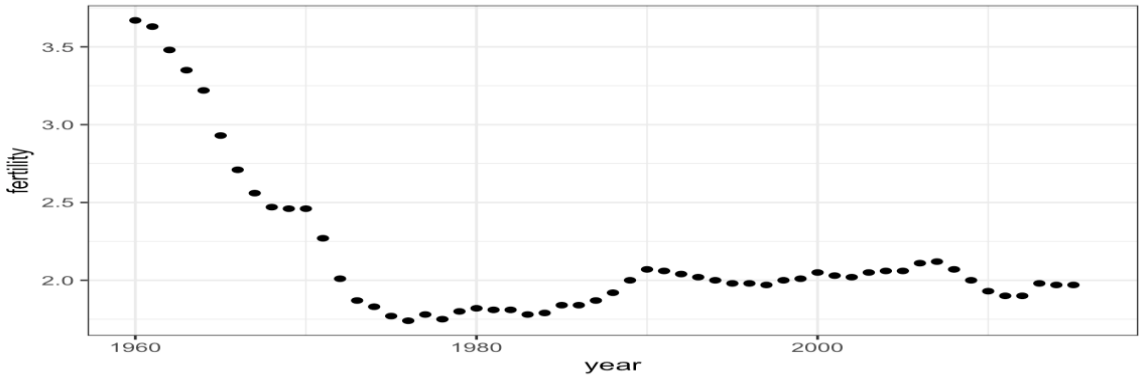
In the plot above, we have to pay special attention to the range to notice that the plot on the right has a larger life expectancy.

5. Time series plots

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

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

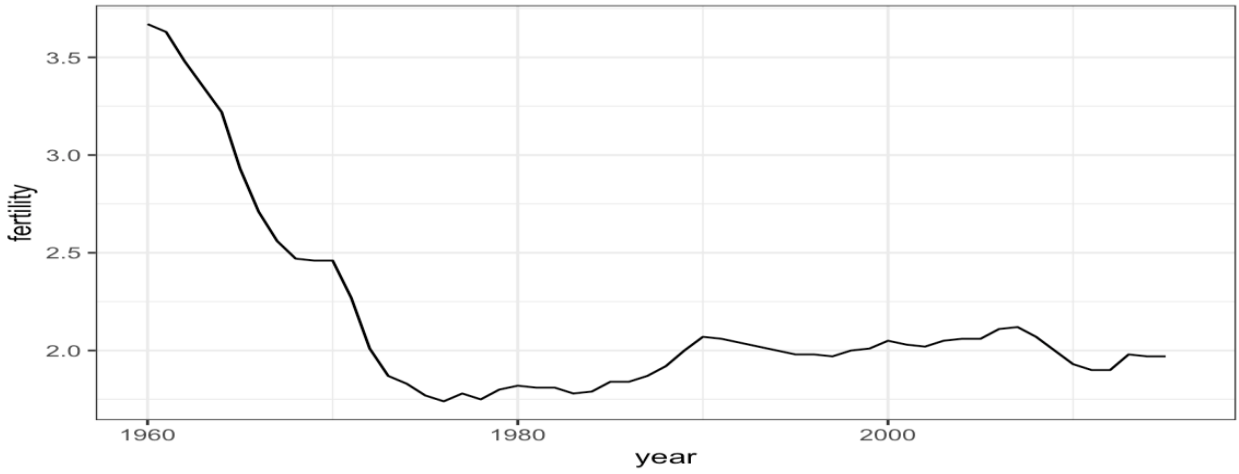
```
gapminder |>
  filter(country == "United States") |>
  ggplot(aes(year, fertility)) +
  geom_point()
```



We see that the trend is not linear at all. Instead there is sharp drop during the 1960s and 1970s to below 2. Then the trend comes back to 2 and stabilizes during the 1990s.

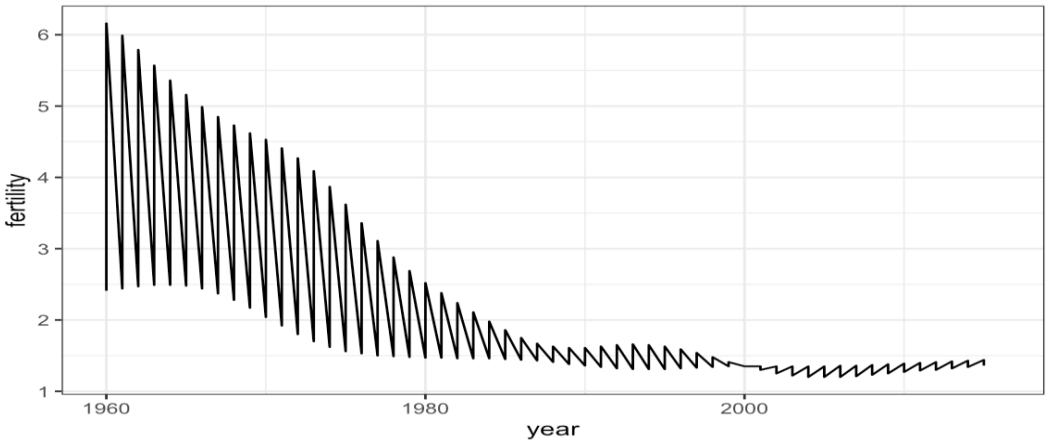
When the points are regularly and densely spaced, as they are here, we create curves by joining the points with lines, to convey that these data are from a single series, here a country. To do this, we use the `geom_line` function instead of `geom_point`.

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



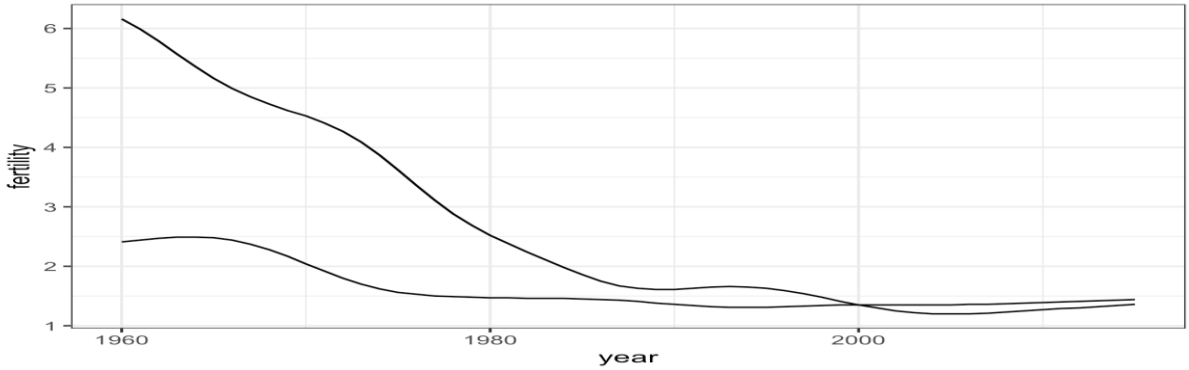
This is particularly helpful when we look at two countries. If we subset the data to include two countries, one from Europe and one from Asia, then adapt the code above:

```
countries <- c("South Korea","Germany")

gapminder |> filter(country %in% countries) |>
  ggplot(aes(year,fertility)) +
  geom_line()
```



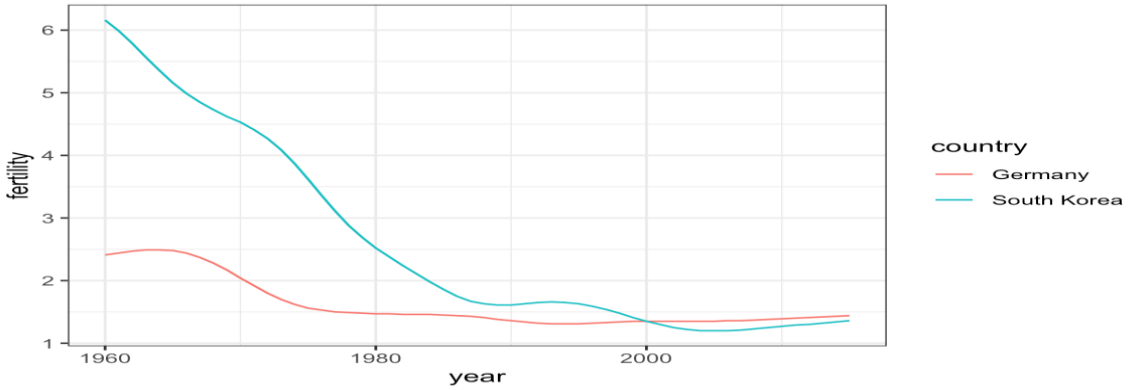
Unfortunately, this is **not** the plot that we want. Rather than a line for each country, the points for both countries are joined. This is actually expected since we have not told `ggplot` anything about wanting two separate lines. To let `ggplot` know that there are two curves that need to be made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

```
countries <- c("South Korea","Germany")

gapminder |> filter(country %in% countries & !is.na(fertility)) |>
  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 & !is.na(fertility)) |>
  ggplot(aes(year,fertility, col = country)) +
  geom_line()
```



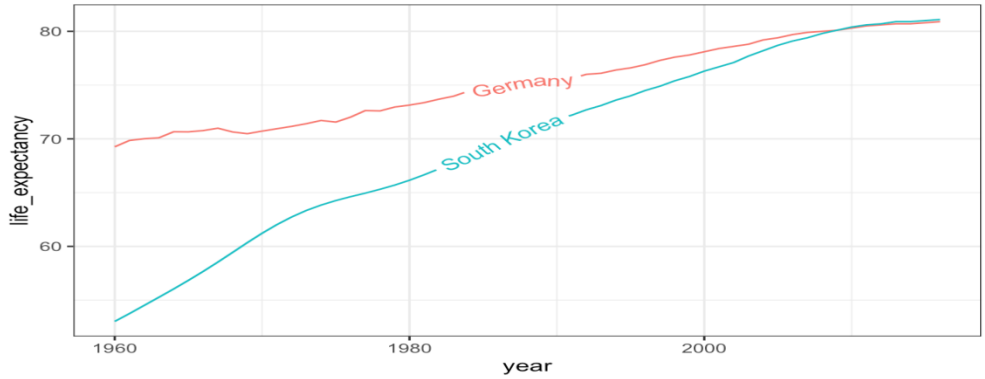
The plot clearly shows how South Korea's fertility rate dropped drastically during the 1960s and 1970s, and by 1990 had a similar rate to that of Germany.

6 Labels instead of legends

For trend plots we recommend labeling the lines rather than using legends since 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 `geomtextpath` package. We define a data table with the label locations and then use a second mapping just for these labels:

```
library(geomtextpath)

gapminder |>
  filter(country %in% countries) |>
  ggplot(aes(year, life_expectancy, col = country, label = country)) +
  geom_textpath() +
  theme(legend.position = "none")
```

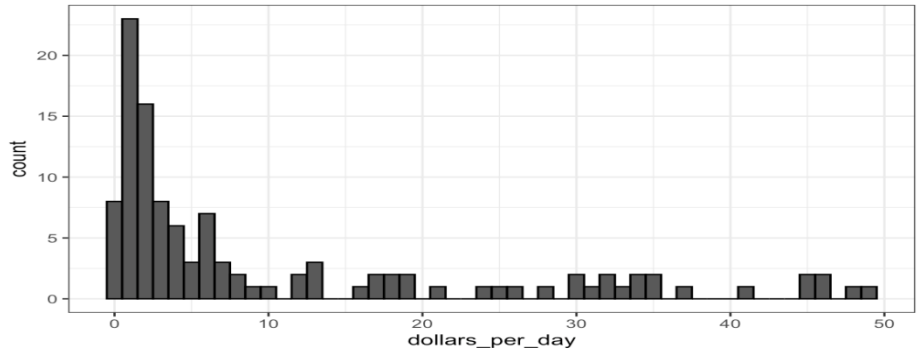


The plot clearly shows how an improvement in life expectancy followed the drops in fertility rates. In 1960, Germans lived 15 years longer than South Koreans, although by 2010 the gap is completely closed. It exemplifies the improvement that many non-western countries have achieved in the last 40 years.

7 Log transformation

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")
```



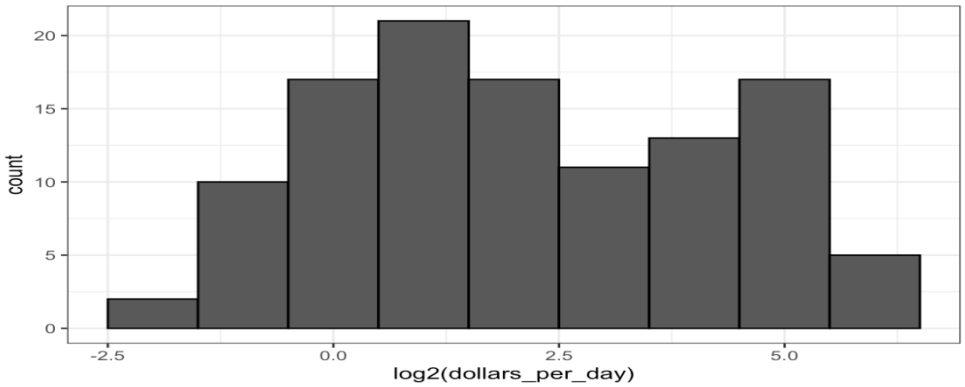
We use the `color = "black"` argument to draw a boundary and clearly distinguish the bins.

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 convert 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 transform:

```
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.

III.What are Data Visualization Tools?

Data Visualization Tools are software platforms that provide information in a visual format such as a **graph**, **chart**, etc to make it easily understandable and usable. Data Visualization tools are so popular as they allow analysts and statisticians to create visual data models easily according to their specifications by conveniently providing an interface, database connections,

Top Data Visualization Tools

1. Tableau

Tableau is a data visualization tool that can be used by data analysts, scientists, statisticians, etc. to visualize the data and get a clear opinion based on the data analysis. Tableau is very famous as it can take in data and produce the required data visualization output in a very short time. And it can do this while providing the highest level of security with a guarantee to handle security issues as soon as they arise or are found by users.

Tableau also allows its users to prepare, clean, and format their data and then create data visualizations to obtain actionable insights that can be shared with other users. Tableau is available for individual data analysts or at scale for business teams and organizations. It provides a 14-day free trial followed by the paid version.

2. Looker

Looker is a data visualization tool that can go in-depth into the data and analyze it to obtain useful insights. It provides real-time dashboards of the data for more in-depth analysis so that businesses can make instant decisions based on the data visualizations obtained. Looker also provides connections with more than 50 SQL-supported dialects so you can connect to multiple databases without any issues.

3. Zoho Analytics

Zoho Analytics is a Business Intelligence and Data Analytics software that can help you create wonderful-looking data visualizations based on your data in a few minutes. You can obtain data from multiple sources and mesh it together to create multidimensional data visualizations that allow you to view your business data across departments. In case you have any questions, you can use Zia which is a smart assistant created using artificial intelligence, machine learning, and natural language processing.

Zoho Analytics allows you to share or publish your reports with your colleagues and add comments or engage in conversations as required. You can export

4. Sisense

Sisense is a business intelligence-based data visualization system and it provides various tools that allow data analysts to simplify complex data and obtain insights for their organization and outsiders. Sisense believes that eventually, every company will be a data-driven company and every product will be related to data in some way. Therefore it tries its best to provide various data analytics tools to business teams and data analytics so that they can help make their companies the data-driven companies of the future.

5. IBM Cognos Analytics

IBM Cognos Analytics is an Artificial Intelligence-based business intelligence platform that supports data analytics among other things. You can visualize as well as analyze your data and share actionable insights with anyone in your organization. Even if you have limited or no knowledge about data analytics, you can use IBM Cognos Analytics easily as it interprets the data for you and presents you with actionable insights in plain language.

6. Qlik Sense

Qlik Sense is a data visualization platform that helps companies to become data-driven enterprises by providing an associative data analytics engine, sophisticated Artificial Intelligence system, and scalable multi-cloud architecture that allows you to deploy any combination of SaaS, on-premises, or a private cloud.

You can easily combine, load, visualize, and explore your data on Qlik Sense, no matter its size. All the data charts, tables, and other visualizations are interactive and instantly update themselves according to the current data context. The Qlik Sense AI can even provide you with data insights and help you create analytics using just drag and drop. You can try Qlik Sense Business for free for 30 days and then move on to a paid version.

7. Domo

Domo is a business intelligence model that contains multiple data visualization tools that provide a consolidated platform where you can perform data analysis and then create interactive data visualizations that allow other people to easily understand your

data conclusions. You can combine cards, text, and images in the Domo dashboard so that you can guide other people through the data while telling a data story as they go.

8. Microsoft Power BI

Microsoft Power BI is a Data Visualization platform focused on creating a data-driven business intelligence culture in all companies today. To fulfill this, it offers self-service analytics tools that can be used to analyze, aggregate, and share data in a meaningful fashion.

9. Klipfolio

Klipfolio is a Canadian business intelligence company that provides one of the best data visualization tools. You can access your data from hundreds of different data sources like spreadsheets, databases, files, and web services applications by using connectors. Klipfolio also allows you to create custom drag-and-drop data visualizations wherein you can choose from different options like charts, graphs, scatter plots, etc.

10. SAP Analytics Cloud

SAP Analytics Cloud uses business intelligence and data analytics capabilities to help you evaluate your data and create visualizations in order to predict business outcomes. It also provides you with the latest modeling tools that help you by alerting you of possible errors in the data and categorizing different data measures and dimensions. SAP Analytics Cloud also suggests Smart Transformations to the data that lead to enhanced visualizations.

.

11. Yellowfin

Yellowfin is a worldwide famous analytics and business software vendor that has a well-suited automation product that is specially created for people who have to take decisions within a short period of time. This is an easy-to-use data visualization tool that allows people to understand things and act according to them in the form of collaboration, data storytelling, and stunning action-based dashboards.

IV.Some Inspiring Data Science Projects

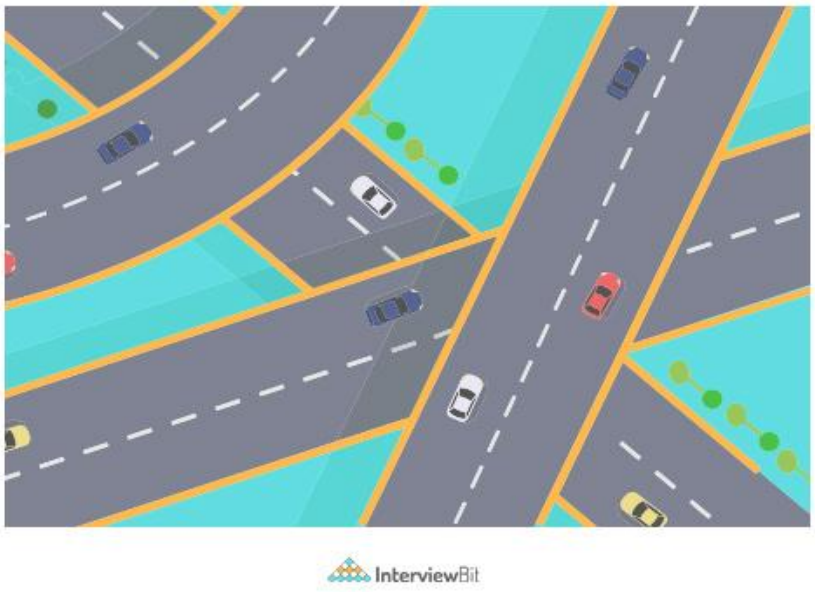
1. Fake News Detection Using Python

Fake news do not require any introduction. It is very much easy to spread all the fake information in today's all-connected world across the internet. Fake news is sometimes transmitted through the internet by some unauthorised sources, which creates issues for the targeted person and it makes them panic and leads to even violence. To combat the spread of fake news, it's critical to determine the information's legitimacy, which this **Data Science project** can help with.

2. Data Science Project on Detecting Forest Fire

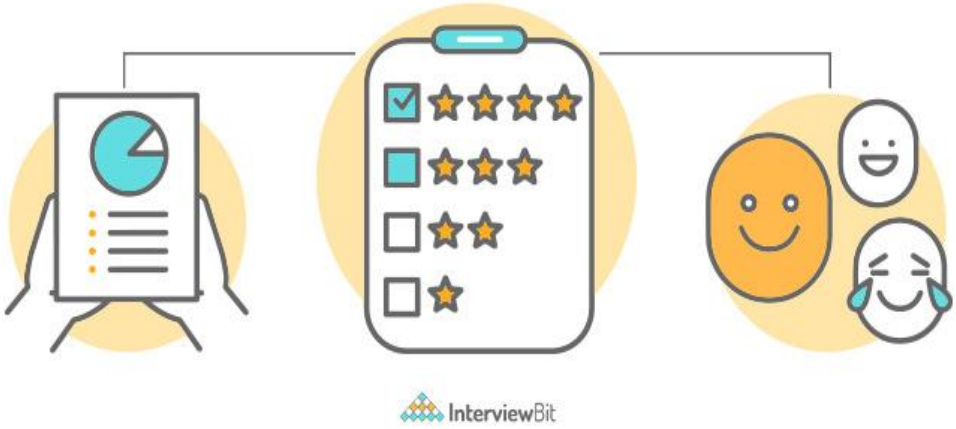
Developing a project for identifying the forest fire and wildfire system is an alternatively good example to exhibit one's skills in Data **Scienc**e. The forest fire or wildfire is an uncontrollable fire that develops in a forest. All the forest fir will create havoc during weekends on the animal habitat, surrounding environment and human property. k-means clustering can be used for the identification of the crucial hotspots during forest fire and to reduce the severity , to regulate them and even to predict the behaviour of the wildfire. This is advantageous for allocating the required resources. To enhance the model's accuracy,

3. Detection of Road Lane Lines



A Live Lane-Line Detection Systems built-in Python language is another Data Science project idea for beginners. A human driver receives lane detecting instruction from lines placed on the road in this project. The lines placed on the roads indicate where the lanes are located for human driving. It also refers to the vehicle's steering direction. This application is crucial for the development of self-driving cars. This application for the Data Science Project is critical for the development of self-driving cars.

4. Project on Sentimental Analysis



The act of evaluating words to determine sentiments and opinions that may be positive or negative in polarity is known as sentimental analysis. This is a sort of categorization in which the classifications are either binary (optimistic or pessimistic) or multiple (happy, angry, sad, disgusted, etc.). The project is written R Language,

5. Project on Influences of Climatic Pattern on the food chain supply globally

The abnormalities and changes occurring in the climate very often are the main challenges impressed on the environment that needs to be taken care of. These environmental changes will affect the human beings on earth. This Data Science Project makes an attempt to analyse the changes in the food production globally that occurs due to change in climatic conditions. The main purpose of this study is to evaluate the consequences of climatic changes on primary agricultural yields. This project will evaluate all the effects related to change in temperature and rainfall pattern. The amount of carbon dioxide that impacts plant development and the uncertainties in climate change will next be considered. As a result, data representations will be the primary focus of this project. It will also assess productivity across different locations and geographical regions

V.Create your own Visualization of a Complex dataset

----ACTIVITY ---

--------By-----

------U-----

Or Ask me……