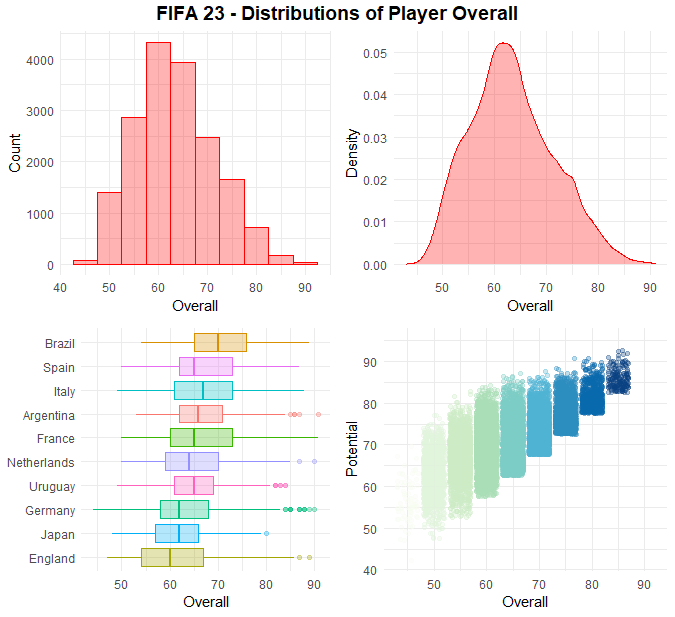
**How To R: Visualizing Distributions**

Distributions are a very important thing to understand during any EDA process. In this article we will look into multiple different ways to visualize distributions and how to create those visuals in R.

Visualizing Distributions



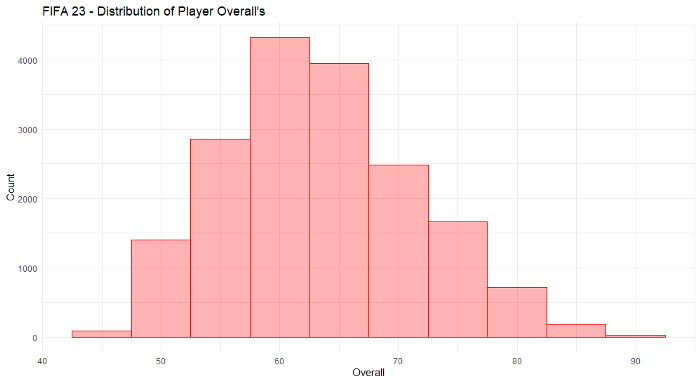
For this article I will be using the [FIFA OFFICIAL DATASET](https://www.kaggle.com/datasets/bryanb/fifa-player-stats-database) from Kaggle and the R package ggplot2 to create the visualizations.

I will be going over the following with code to show how each was created:

1. Histogram
2. Box Plots
3. Density Charts
4. Ridgeline Charts
5. Heatmap
6. Jitter Plots

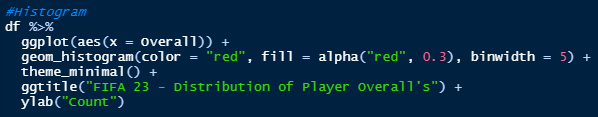
**1. Histogram**

*Basic Histogram*



With the histogram you are able to easily see which overall’s are attributed to the most players. The distribution of this data seems to be close to a normal distribution.

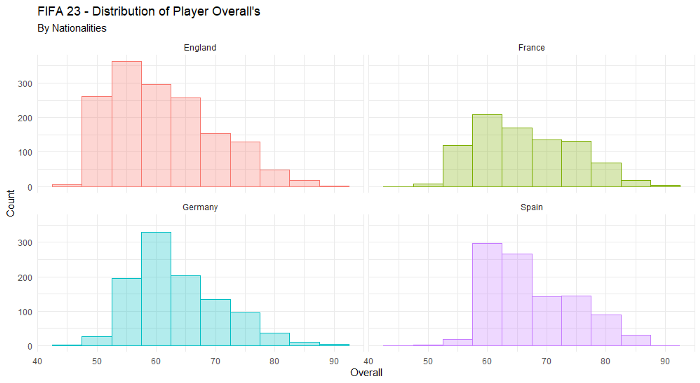
Code used for the chart above.



Things to note:

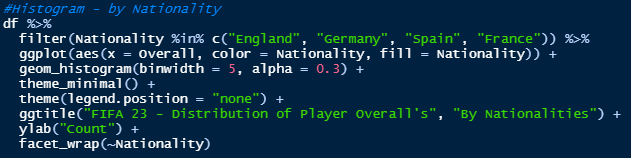
* geom\_histogram is the function used to create the histogram. The binwidth is defaulted but you can change this, as I did by using binwidth. You can also use bins instead of binwidth, if you wanted to choose how many bars would show up. A binwidth of 5 would separate the data into 0–5 overall, 5–10, 10–15, etc. and make up as many bars needed to fill the whole dataset. A bins of 5 on the other hand would split the data into 5 even groups, so for a dataset going from 0 to 100 then the 5 bars would each have 20 to it (0–20, 20–40, etc).
* The color of the histogram can also be altered within the geom\_histogram function. The color relates to the border on the outside of the bars, while fill is the inside of the bars.

*Histogram by Group*



Here I split the histogram into the player nationalities for the 4 nations with the highest count of players. Here we can see how each one of these nations are distributed.

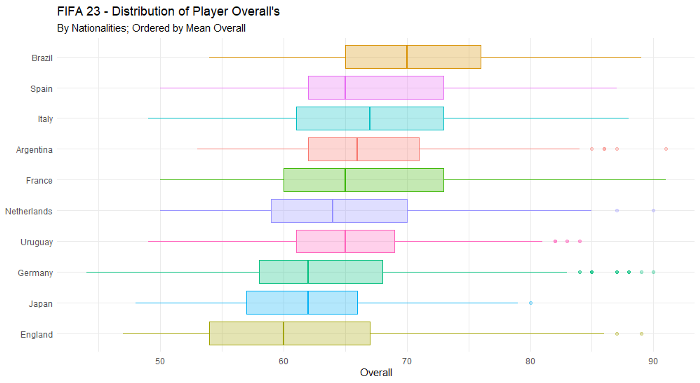
Code used for chart above.



Things to note:

* I made the color tied to each of the nationalities within the ggplot function. This adds a legend, but since it is clear what we are looking at I removed the legend using the theme function.
* I split the histogram into 4 different histograms, 1 for each nationality, by using the facet\_wrap function at the bottom of the code.

**2. Box Plots**



With this layout we can see the distribution of many more nationalities than we did with the histogram, now showing the top 10 nationalities by player count. I also sorted the nations seen here by their mean Overall.

Code used for the graph above.

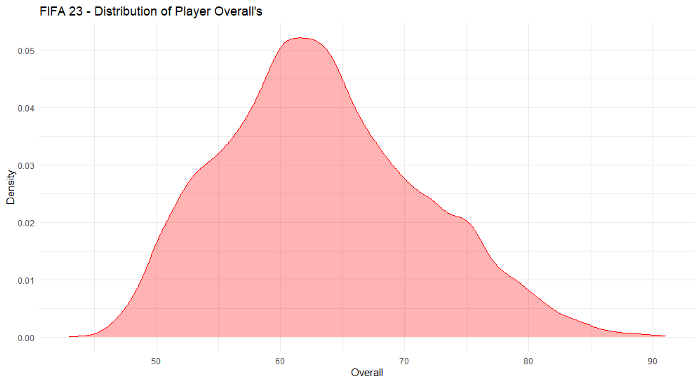


Things to note:

* I found the mean of each nationalities overall by performing a group by on nationality prior to creating the ggplot, creating a new field “mean\_OVR” in the process.
* I set the order of the nationalities by mean\_OVR within the ggplot function, under the y = section.

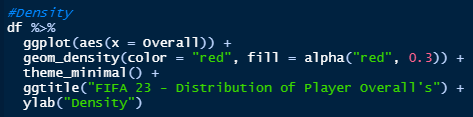
**3. Density Charts**

*Basic Density Chart*



The density chart is similar to the histogram, however it shows the distribution without the needs of bins or binwidths and instead by the density seen at each point.

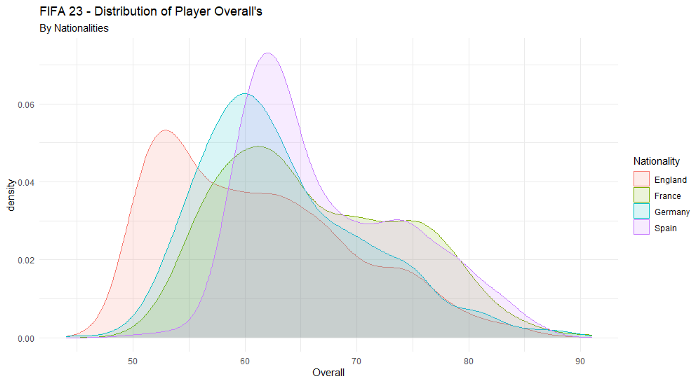
Code used in graph above.



Things to note:

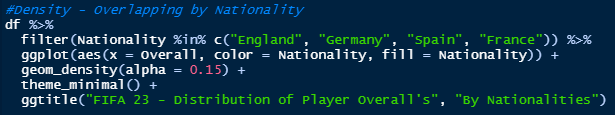
* To create the density chart, the geom\_density function was used

*Overlapping Density Chart*



In this variation, we have the top 4 nationalities by player count shown. Their density plots are overlapping each other and as we do not have too many nationalities this chart works for us to compare them. We can see that England has the lowest spike for overall at around 55, then Spain has the largest spike and also the tallest one, at around 65.

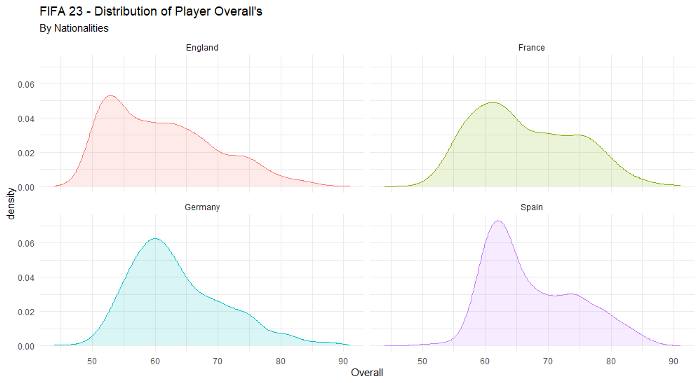
Code used in the graph above.



Things to note:

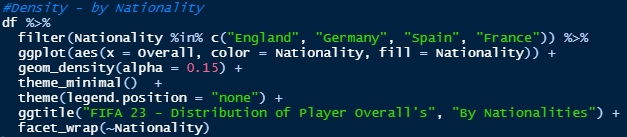
* The groupings by nationality occurred simply by adding color and fill = Nationality in the ggplot function.

*Density Chart by Group*

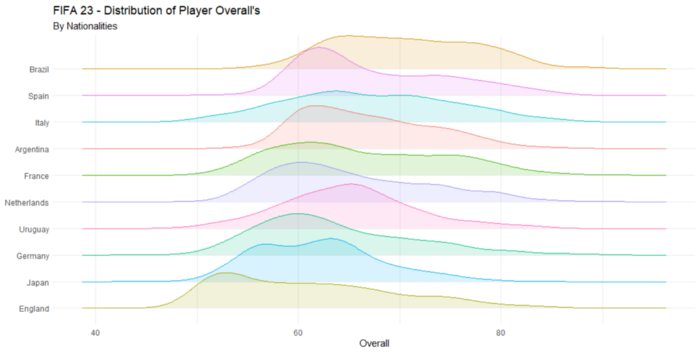


This visual is similar to how we looked at the histogram in the first section. This would be useful instead of the overlapping density chart in case there were many different groupings and it was too difficult to compare each one.

Code used in the graph above.

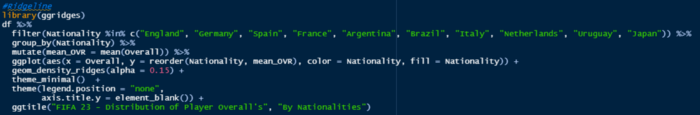


**4. Ridgeline Charts**



The ridgeline plot is an easy way to show the density distribution for each group at once. I sorted this set by mean overall again.

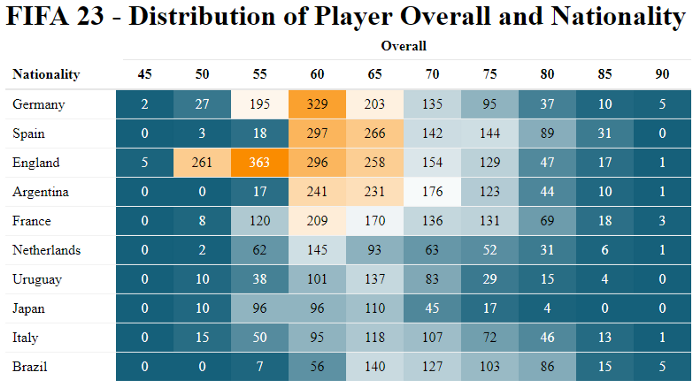
Code for graph above.



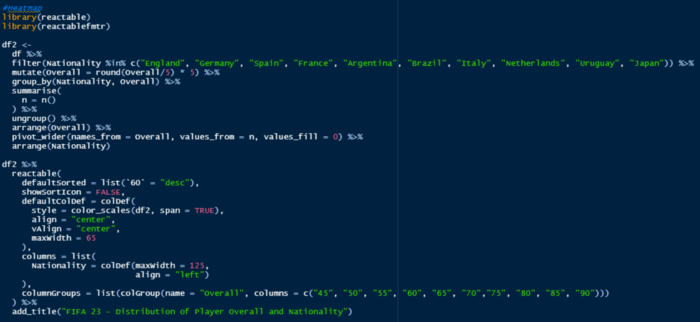
Things to note:

* To create this chart, you will need to also install the ggridges library.
* You will use the function geom\_density\_ridges in place of the geom\_density function. The rest will be handled the same way.

**5. Heatmap**



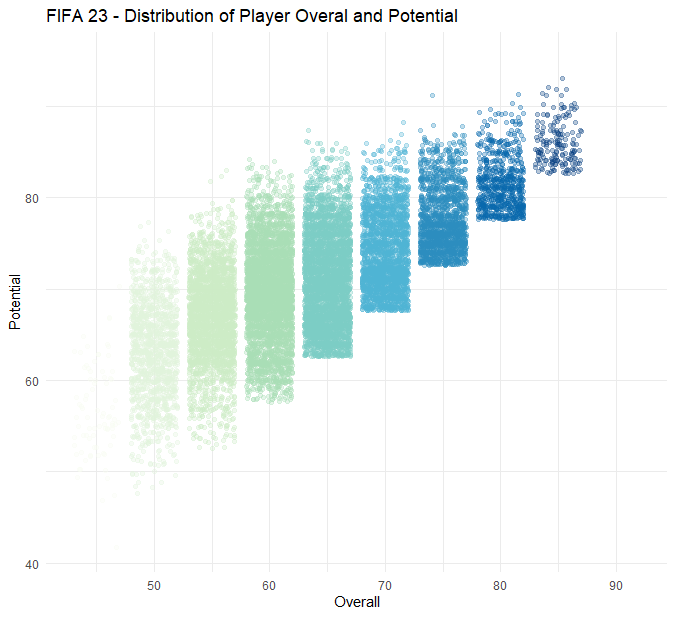
Code used for graph above.



Things to note:

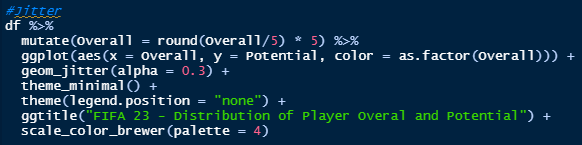
* For this heatmap I used the reactable and reactablefmtr packages.
* Prior to placing it into the reactable function, I needed to set up the data.

**6. Jitter Plots**



This chart shows each point for every overall grouped to the nearest 5, comparing the overall to the potential of that player. The jitter spaces them just a bit out on the x-axis so that it is easier to see how many points are on top of each other.

Code for graph above.



Things to note:

* You have to set up ggplot in the same way you would set up a geom\_point (scatter plot) but instead of using that function you would instead use geom\_jitter.
* I changed the color palette with scale\_color\_brewer. This makes it easy to select different color palettes.