GGPLOT2 visualization beyond histograms, scatterplots, and bar charts

Roberto Romo

12/2/2017

Introduction and Motivation

In this course we have learned different visualization packages like 'ggplot2' and 'ggvis' on top of the stock plotting methods. However, we have primarily restricted our visualization to barcharts, histograms, and scatterplots. In an effort to expand our visualization methods, we will be looking at different plot types from the 'ggplot2' package that can help us convey what we want to in different and better ways. (This post assume knowledge of the 'ggplot2' package and will not provide instruction on it beyond provided methods and code.) For example, instead of just having the usual histogram plot frequency we want it to be more telling of a distribution, as we will see in the next section. Similarly, we may want to explore different visualization methods such as a correlogram that visualizes data in correlational matrices to answer determine if the data is random or if the data is related to other observations. Or finally, we may tend towards visualization that isn't as informative but is creative and beautiful to look at.

In this post we will be exploring different visualization methods via the 'ggplot2', 'ggcorrplot' packages with improved histograms, correlograms, violin plots, density plots, and divering bars. We will be using the R provided "mtcars" dat and "cars.csv" data.

```
library(ggplot2)
library(ggcorrplot)
data(mtcars)
dat <- read.csv("http://web.pdx.edu/~gerbing/data/cars.csv")
```

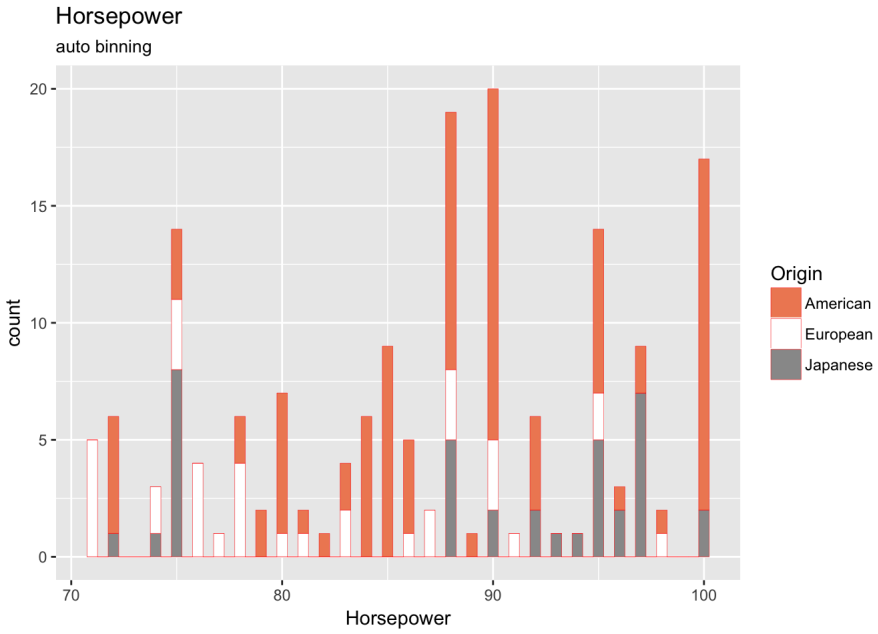
Histograms

Histograms are best suited for distribution; when you want to see how data points are distributed. In this course we have create histograms that demonstrate the frequency of data points but we did not construct histograms that seperated data points by class. The typical distribution histograms inform of spreads and their different shapes (for example normal or right-skewed). But say for example, that you have a dataset of cars created by 3 different producers: European, Japanese and American and say that you wanted to visualize the distribution of the producer cars' horsepower ranging from roughly 70 to 100 horsepower. Now, we want to seperate these cars by their producer or their class and potentially we can have a bar per producer per horsepower but that would make our histogram crowded, repetitive, and difficult to look at. Instead, we can "stack" or fill our bars with their Origin so that there is only one bar per horsepower where the count of the individual origin is represent by a color (here orange, white and grey). We will use the ggplot geom_histogram() function to visualize this distribution.

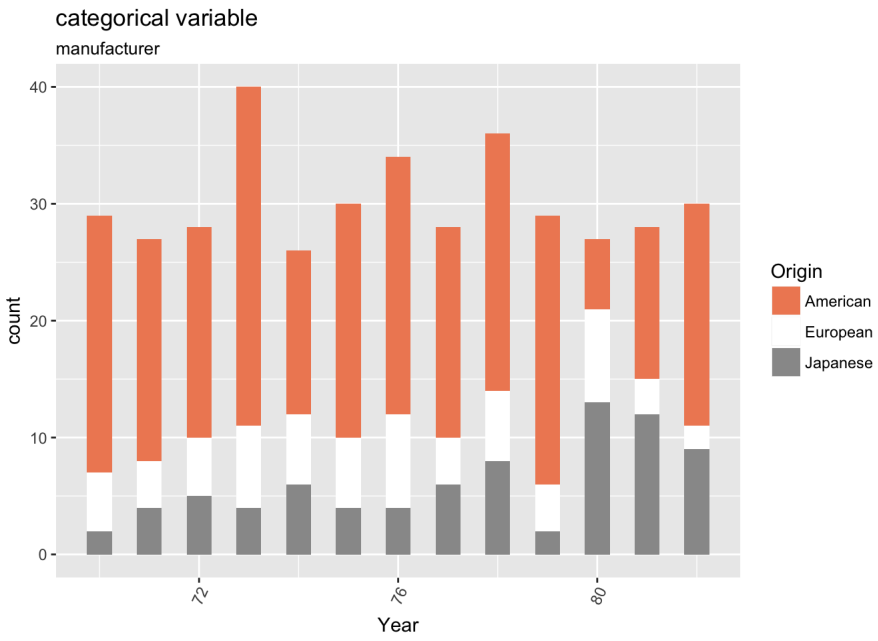
What makes this visualization powerful is that it is easy to look at, it isn't crowded, and it is intuitive. In this case it has helped that the American producers have created more cars (as we see the orange that represents the American producers taking over the top of the bars). Differentiating with the colors in a single bar also helps displays important findings, for example we can observe that there isn't any car produced by the Japanese with horsepower from about ~77 to ~87 horsepower. Another important aspect of this histogram is that narrowing down a bar per horsepower truly is easy to look at it instead; for example dominant classes are easy to determine along the least dominant because we can easily determine which colors dominante the plot. Another example of this simple and clean yet powerful visualization is the plot that shows the distribution of cars made per year (from 1970 to 1990) from Japanese, American, and European producers. Although the data set has made it easy for us to see how orange (American) dominates while the (Japanese) produced less cars, this histogram with classes reveals information in a way that doesnt require much obersvation to understand. Had there been more of a balanced distribution or one where it not a single class dominated, the color spread on the bars would have informed that.

```
newDat <- dat[dat$Horsepower <= 100 & dat$Horsepower > 70,]
g <- ggplot(newDat, aes(Horsepower)) + scale_fill_brewer(palette = "RdGy")

g + geom_histogram(aes(fill=Origin),
                   binwidth = .5,
                   col="red",
                   size=.1) +
  labs(title="Horsepower",
       subtitle="auto binning")
```




```
g <- ggplot(dat, aes(Year)) + scale_fill_brewer(palette = "RdGy")
g + geom_bar(aes(fill=Origin), width = 0.5) +
  theme(axis.text.x = element_text(angle=65, vjust=0.6)) +
  labs(title="categorical variable",
       subtitle="manufacturer")
```



Correlograms

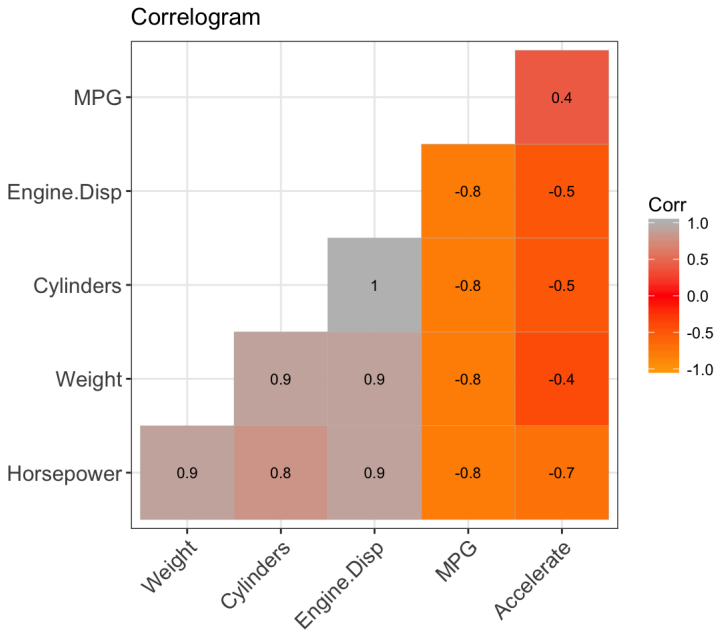
Now suppose you wanted to show the correlation between different data points of a dataset. It is by no mistake that in this course we have resorted to the use of scatterplots to understand the nature between two variables. The scatterplot is intuitive, it is simple, and it does just what it's supposed to do; show the correlation between an x-axis data point and its y-axis counterpart. But what about when we want to examine the correlation between more than two variables, say multiple continuous variable (data points) in our data set. CORRELOGRAMS! Here, we will use the 'ggcorrplot' to visualize the correlation between the data points MPG, Engine Displacement, Cylinders, Weight, and Horsepower of the cars produced. We will forgo the classes in this example because our ultimate goal is to understand the correlation of these data points as a whole for cars not per producer.

We first calculate the correlation of the given data points and insert it into the the ggcorrplot() function to visualize the nature of the intersection of these data points. We can see to the right of the correlogram that the greys indicate a ture correlation of 1.0 while the red indicates 0 correlation and a tangerine orange indicates a -1.0 correlation (note these correlations are rounded for the purpsoe of this example). Additionally, we have ordered these correlations so that the higher correlations tend to the left and the lower correlations tend to the right. This ordering produces a visualization that is clean and easy to understand. For example, a car that weighs more will be in need of more horsepower than a car that weighs less (hence the correlation is 0.9), similarly MPG is not an indicator of horsepower as there is not true correlation (hence correlation of -0.8).

With the correlogram we can display the correlation of multiple continuous data points and present all of them at once.

```
newDat <- dat[, 2:7]

corr <- round(cor(newDat), 1)
ggcorrplot(corr, hc.order = TRUE,
           type = "lower",
           lab = TRUE,
           lab_size = 3,
           method="square",
           colors = c("orange", "red", "grey"),
           title="Correlogram",
           ggtheme=theme_bw)
```



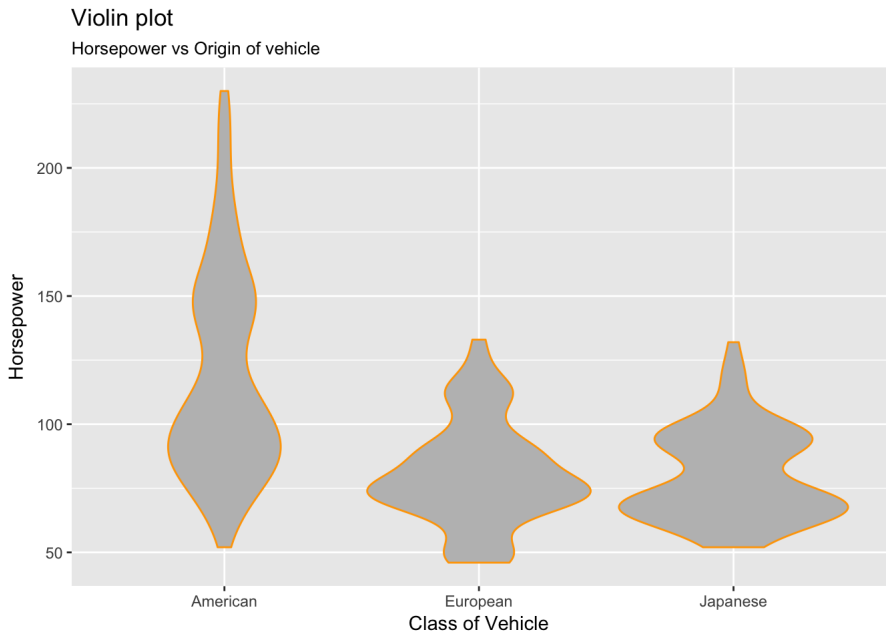
Violin

We then present the violin plot. The violin plot is the histogram's sibling in that it is used to display density. The violin plot can be plotted with 'ggplot2' with the geom_violin() function and although this version of the violin plot isn't very informative or as intuitive as the displayment of the histogram it is a "fun" plot to look at and create. Here you can find further information on the violin plot and it's components such as how the Median or confidence interval is represented.

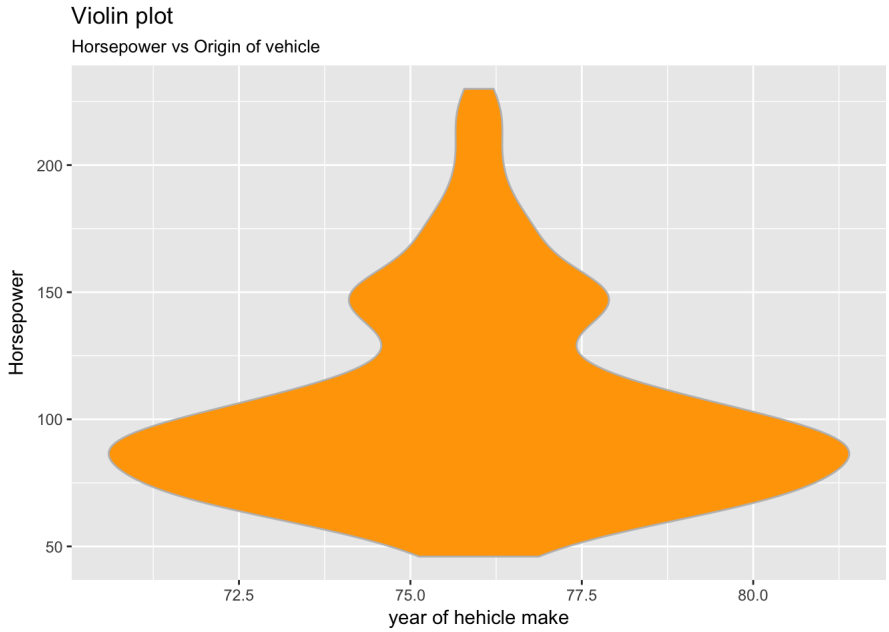
I present the violin plot as a means to explore plots and graphs that are unusual and creative but nonetheless display our message (whether less powerful or not, it is up to you to decide). For the most part, the violin plot is a combination of the box plot which we have seen in this course and the density plot (which we will also learn about). The violin plot is one of those plots that tend towards artforms. In the examples below, for the first plot we plot the density of horsepower of cars per the 3 producers and the density of horsepower of cars per the year of its making. What we ultimate visualize is the how many cars produced by each producers with certain horsepower and although we bypass exact features of this plot (for example mean or confidence intervals), the sleek and simple plot reveals valuable information. We determine that the American producers produced vehicles of higher horsepower than it's European and Japanese counterparts by a lot while the European and Japanese producers produced cars in the lower horsepowers of 55 to ~85. With the odd shapes of each producers, density is literal; it takes shape by density of its horsepower and that spatial presence informs us of that density.

The takeway from this plot is not that it is formal, but on the contrary that data visualizaton does not always have to be formal or follow a set of rules. It is to encourage you to visualize data in a ways you best feel conveys your message and to extend beyond the usual convention or the usaul histogram or scatterplot. That is not to undermine the power of these plots, but to encourage you to seek different means to explore, present, and understand data. For example, albeit the second plot displayed is more for the fun of it than for actual display of information, it still displays what it indends. We can conlcude from this plot that the years were the cars were produced with more horsepower where from the 75's to the 77.5's (in this dataset.)

```
g <- ggplot(dat, aes(Origin, Horsepower))
g + geom_violin(color = "orange", fill = "grey") +
  labs(title="Violin plot",
       subtitle="Horsepower vs Origin of vehicle",
       x="Class of Vehicle",
       y="Horsepower")
```




```
g <- ggplot(dat, aes(Year, Horsepower))
g + geom_violin(color = "grey", fill = "orange") +
  labs(title="Violin plot",
       subtitle="Horsepower vs Origin of vehicle",
       x="year of hehicle make",
       y="Horsepower")
```

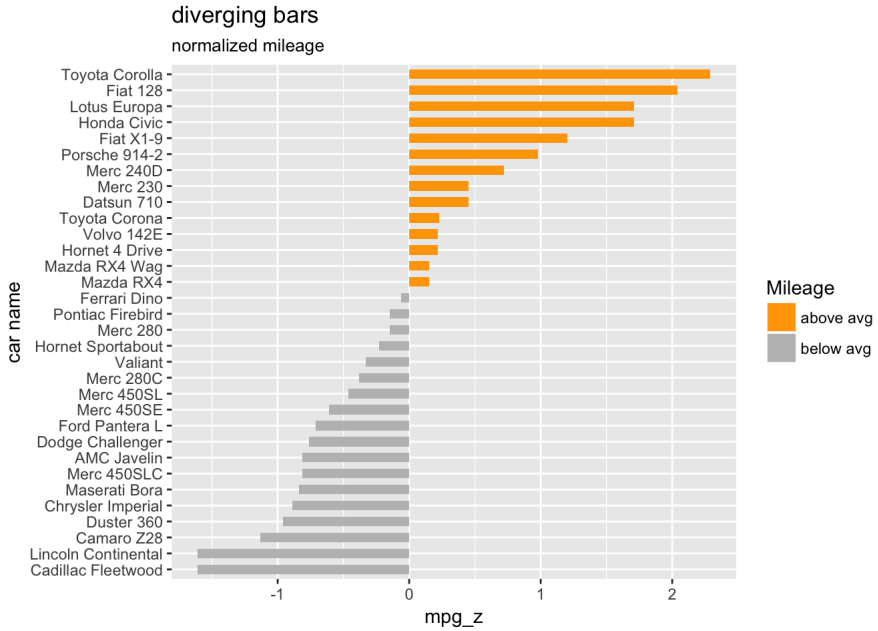


Diverging Bars

Diverging bars are intended to compare varition in data points with respect to a category or reference. Although a divering bar may look just like the histograms above with switched orientation it is stacked horizontally so that “negative” variation is place on the left side and “positive” on the right side. Diverging bars are handy when you want to compare a column of a dataset amongst all rows but have a threshold that determines for example what is “good/positive” and “bad/negative”. To plot the diverging bars we use the provided geom_bar() function from the ‘ggplot2’ package and within it set stat=‘identity’ and provide an x and y for aes(). In this example we compare then ormalized mileage (by computing the z score shown below) of different cars where we identigy a normalized mileage below 0 as “below average” and those above 0 as “above average”.

The divering bar plot is able to take on “positive” and “negative” values to show the deviation of data points (in this case with respect to the z score). What makes this visualization powerful (albeit the provide example in comparison to a divering bar plot like this) is the idea of negativeness and positiveness in that we perceive generally an item placed on the right to be above or ahead than an item that is placed to the left of that item. Additionally, we have in this plot, following the metric that placing an item above is better and good than an item placed at the bottom, that the cars with above average mileage are placed at the top and the cars with below average mileage are placed at the bottom. The hierarchical structure in the divering bar plot, not only in that it presents data left to right but top to bottom, visualizes clean, intuitive, and easy to read data (as you can easily pinpoint and compare the deviation of cars in the example by the orientation of the bars and their color).

```
mtcars$`car name` <- rownames(mtcars)
mtcars$mpg_z <- round((mtcars$mpg - mean(mtcars$mpg))/sd(mtcars$mpg), 2)
mtcars$mpg_type <- ifelse(mtcars$mpg_z < 0, "below", "above")
mtcars <- mtcars[order(mtcars$mpg_z), ]
mtcars$`car name` <- factor(mtcars$`car name`, levels = mtcars$`car name`)
ggplot(mtcars, aes(x=`car name`, y=mpg_z, label=mpg_z)) +
  geom_bar(stat='identity', aes(fill=mpg_type), width=.6)  +
  scale_fill_manual(name="Mileage",
                    labels = c("above avg", "below avg"),
                    values = c("above"="orange", "below"="grey")) +
  labs(subtitle="normalized mileage",
       title= "diverging bars") +
  coord_flip()
```



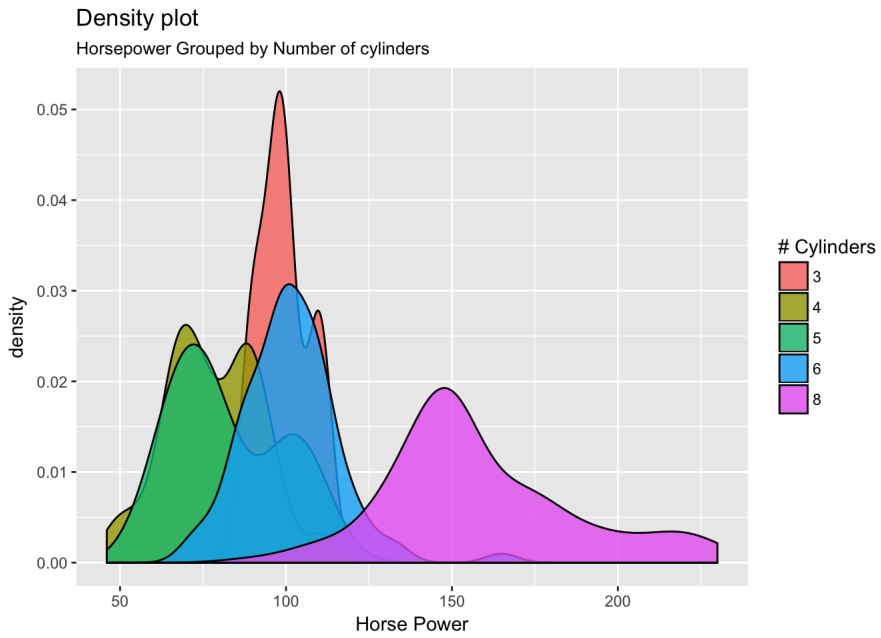
Density Plot

Last but not least we will present the density plot. The density plot like the violin plot is intended to visualize the density of data points, however; with the intent of presenting the change of the density over time (in essence a continuous interval or period of time). In the violin plot we presented the density of different data points as whole but what if we wanted to observe the change in density over time. We will use the geom_density() function from the 'ggplot2' package to visualize the density plot of the cylinder of cars given their horsepwer and then the cylinder of cars over the progression of the 70's to the 80's. In the first plot we visualized the density of the cars' cylinder relative to their horsepower, however; the horsepower is not a continuous period of time or interval. We are adopting the density plot to delivery a finding that doesn't follow the typical usage of a density plot and that is powerful not only because we are still visualizing our intent but because deviation from the norm is good, sometimes.

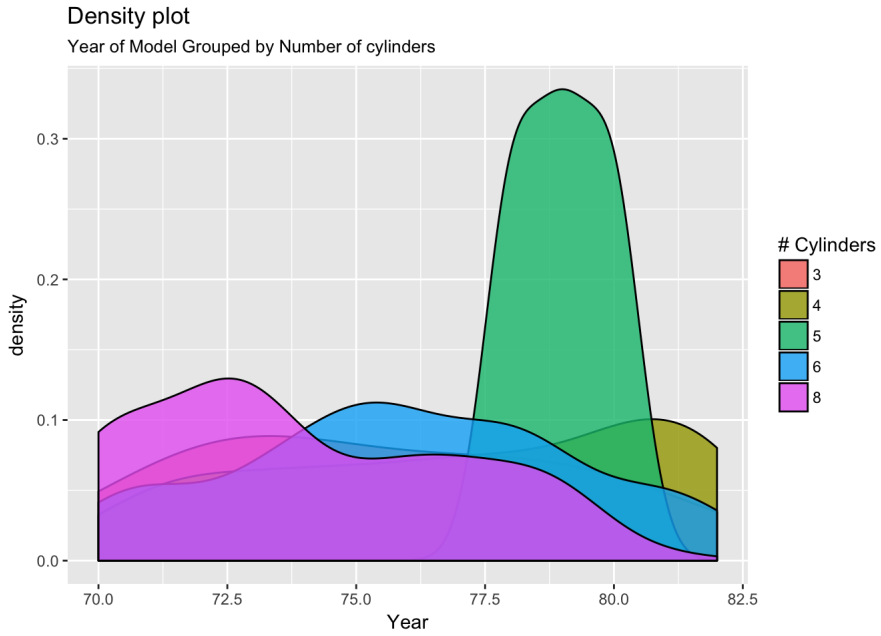
Perhaps a better plot for this finding would have been a violin plot but the message delivered is equally important and presented. We use, here, horsepower as a continuous "interval" and we are able to simulate a progression of (the density) how many cylinders a car has in relation to their horsepower (note however that a car's cylinder does not affect a car's horsepower and in this usage of the density plot we seem to be wanting to show correlation when none exist. The only "correlation" that should be presented in the density plot is the progression of time and how it might affect data points). Perhaps, for this first example, the density plot may have not been the wisest decision (especially due to our choice of data points) but the idea is to encourage you to explore data visualization and trying to adapt to these methods by sometimes following the norm and sometimes deviating.

Finally, our second example demonstrates the density of cars' cylinder over a continuous interval, the years 1970's to the 1980's. The continuous hill-like figures span across the entire plot giving it contiuity and presence. But most importantly,the density plot allows for the different hill-like figures to demonstrate the distribution shape (i.e. normal, skewed left…) For example we see that cars of 5 cylinders are demonstrated in a density shape that is skewed right indicating that it's production was found towards the latter years of that interval while cars of 6 cylinders take presence over the entire interval (demonstrated by a normal-like density shape).

```
g <- ggplot(dat, aes(Horsepower))
g + geom_density(aes(fill=factor(Cylinders)), alpha=0.8) +
    labs(title="Density plot",
         subtitle="Horsepower Grouped by Number of cylinders",
         x="Horse Power",
         fill="# Cylinders")
```




```
g <- ggplot(dat, aes(Year))
g + geom_density(aes(fill=factor(Cylinders)), alpha=0.8) +
    labs(title="Density plot",
         subtitle="Year of Model Grouped by Number of cylinders",
         x="Year",
         fill="# Cylinders")
```



Conclusion and Take-Aways

Data is beautiful but so is data visualization. There are practices and techniques used to convey and present data in a way that reaches our audiences and delivers a message, for example you wouldn't want to use a barchart to demonstrate correlation but you may want to use a bar chart to show ranking. While this post wasn't a tutorial on selecting the right colors or shapes of our plots and graphs, it is a friendly post that introduces to plots and graphs that we didn't see in the course but nonetheless useful to present data in different ways. This post is intended to give some comparison and use for the presented plots and it intends to encourage you to look at other data visualization methods and packages that demonstrate why data is truly beautiful .

Of course, data visualization is limitless and there are many and much more developed techniques that get as complicated (to create) as the datasets they represent but one isn't always able to tell that from how clean and informative they are. Take for example our density plot above, in this post the data used did not require any cleaning or putting together, but in the case that it may have been that way, the outputted plots are very intuitive in nature. We see that, for example, from the years ~1977 to ~1981 cars of 5 cylinders were being built while cars of 8 cylinders were built from 1970 to 1982 (the years limit from 1970 to 1980 in the plot above). This data visualization format is a bit abstract in that it isn't a plot as geometrically put as say our histograms but it is perfect in demonstrating density; or another way to think about is the "bulk" or "presence" of these different data points (in this case cylinders of cars) of our data sets.

What also makes data beautiful is that there is also another side of data visualization that doesn't have a set of rules. Unlike the cleaning of data that for the most part is always has the same process, data visualization is deployed by the "artist" (well more like Data Scientist or Statician or Computer Science or the Sta133 student), well that's probably farfetched. See in art, the artists doesn't worry about the enthusiast or if their artwork is intuitive to other artists, because if the art is "good" it will be understood. With data, you don't have that extend of freedom but as long as your visualization conveys your findings in a way that gives meaning (and maybe something really looks amazing), your audience will think too,

that data is beautiful. Go out and plot your data!

Citations

https://www.statmethods.net/graphs/density.html https://peltiertech.com/diverging-stacked-bar-charts/
https://datavizcatalogue.com/methods/violin_plot.html https://www.statmethods.net/advgraphs/correlograms.html
https://www.rdocumentation.org/packages/Deducer/versions/0.7-9/topics/ggcorplot http://ggplot2.org http://r-statistics.co/Top50-Ggplot2-Visualizations-MasterList-R-Code.html#5.%20Composition
https://raw.githubusercontent.com/vincentarelbundock/Rdatasets/master/csv/datasets/mtcars.csv http://web.pdx.edu/~gerbing/data/cars.csv