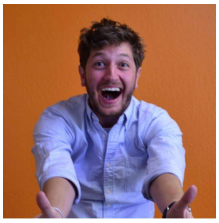
Intro to Data Science

Lesson 04 Notes

Data Visualization

Welcome to Lesson 4



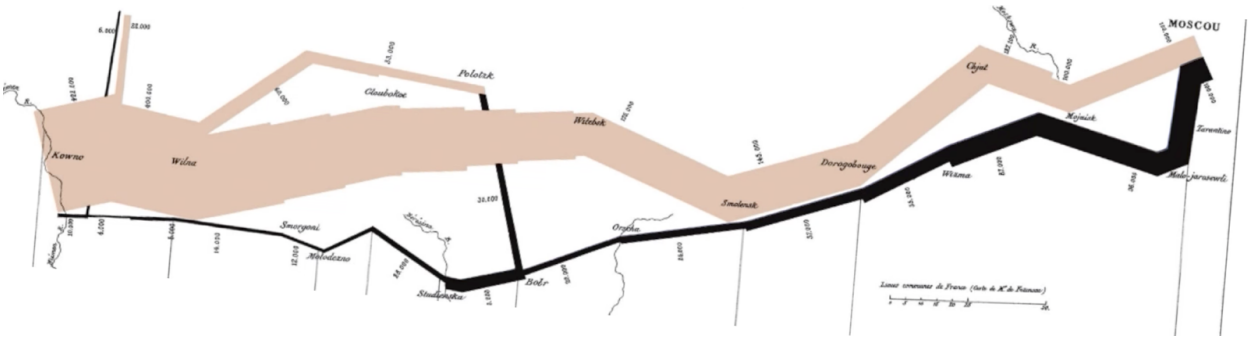
So far in this course, we've discussed how to acquire, munge, and analyze data. These are all really important skill sets for a data scientist to have. However, an equally important skill is the ability to communicate your findings. Sure, this can be done through a presentation or through a blog post. But, one of the most effective ways to communicate your findings is through data visualization. During this lesson we'll be focusing on the basics of data visualization. We'll talk about how to use a variety of visual cues to encode information and also how to make concise and effective visualizations that really communicate what you're trying to say. We'll also go over how to make some of these different types of visualizations in Python. Once you have these skills, you'll be able to take the findings that we've accumulated about the New York City subway system. And really make effective visualizations that communicate these findings either to your friends, or your family. Or, anyone else. All right, why don't we begin?

Effective Information Visualization

To get started, let's answer the question, what is information visualization? Information visualization is the effective communication of complex quantitative ideas. This is usually done through geometry and color. What do we mean by an effective communication? Well, an effective communication is the expression of ideas with clarity, precision, and efficiency. An effective visualization will let you see things that might go unnoticed if you didn't visualize the data. Any data set contains tons of information but you might missed out on trends, behavior patterns, and dependencies if you don't visualize it. Additionally, an effective visualization can highlight certain trends in the data or tell a story to people who might be looking at the visualization. So why don't we take a look at an all time classic visualization of information, Charles Minard's graphical depiction of the terrible fate of Napoleon's army when they marched on Russia.

Napoleons Ill Fated March to Russia

Alright, so, here's a little bit of background on this visualization. Napoleon invaded Russia in 1812, and suffered a number of devastating losses. So Charles Minard created this chart to describe the sheer amount of loss that Napoleon caused in a simple, yet powerful graphic. So if we start at the left of the graphic, we are at the Polish Russian border near the Neman River.



And this flow line shows the size of Napoleon's grand army 422,000 as they invaded Russia of June of 1812. The width of this band indicates the size of the army at each place of the map. In September when the army finally reached Moscow, it had dwindled to 100,000 men. The path of Napoleon's retreat from Moscow is depicted by this lower, darker band which is linked to a temperature scale which is hanging down here. And also these dates that we see at the bottom of the chart. It was incredibly cold, and many froze on the march out of Russia. This chart shows that the crossing of the Berezina River was a disaster. And the army struggled back to Poland with only 10,000 men remaining. Menard made this plot to really highlight the huge amount of loss that was caused by Napoleon's march on Russia. Look at how many men died marching to and from Moscow.

Quiz: What Do You See From This Visualization

As we mentioned earlier, information visualization is the effective communication of complex quantitative ideas. What information do you think is depicted in this visualization? Share your thoughts in the text box below. This quiz is free form so there's no right or wrong answer. I'll share my analysis with you in the following video.

Answer:

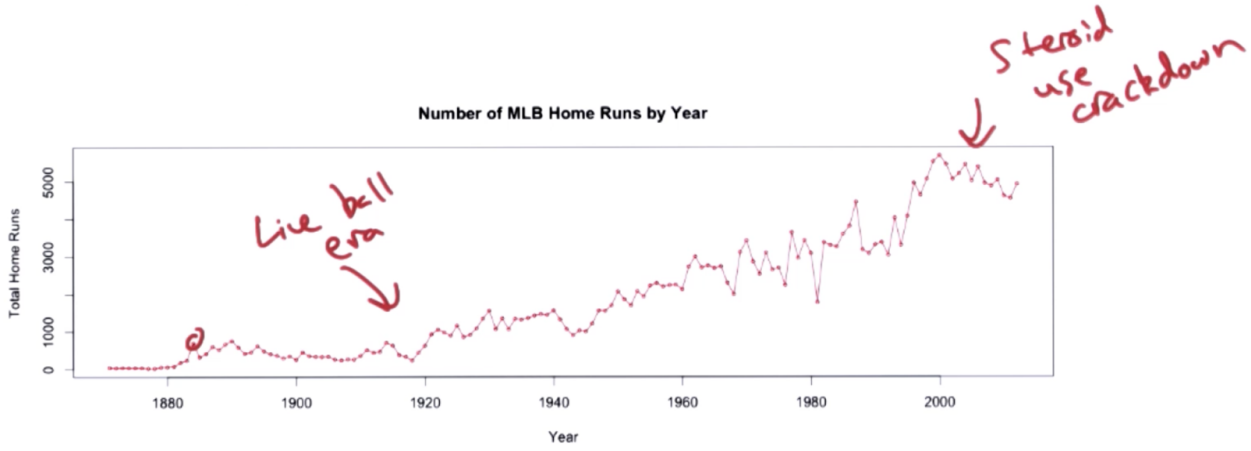
This visualization packs a lot of information into a small area. So, here are some of the details that are communicated. We know the size of Napoleon's army. We know the location of the army at various points in time. The color coding on the march into and out of Moscow tells us the direction of the army's movement. And we're also shown the temperature on various dates during Napoleon's retreat.

Quiz: What Are The Ingredients to Good Infovis

Alright, so we've identified what information this graphic is communicating. But, you could probably imagine a really bad graphic that had all the same information. You might look at it and have no idea what's going on. Or, it might not tell the story as effectively. What makes this graphic so effective? Can we quantify effective information visualization down to a few important ingredients? I'd like to hear your thoughts again. In your perspective, what are the ingredients to good information visualization? This is a free-form question, so there's no right or wrong answer. Type your answer in this text box. I'll share my thoughts with you in the next video.

Answer:

Generally speaking, there are four important ingredients to any information visualization. Visual cues, coordinate systems, scale and data types, and context. Let's use this line chart, which shows total home runs by year in major league baseball to showcase each of these ingredients. So, the first ingredient was visual cues, or visual encoding. This is encoding data with shapes, colors, and sizes. For example, if we look at this chart, we can show that each point shows the number of home runs in a given year. And we can also see that the lines connecting the points give us some sense of the rate of change of number of home runs hit from year to year. There were also coordinate systems. So we need to place our data point somewhere on the chart. A coordinate system gives us a structured space that dictates where the shapes and color should go. This gives meaning to and x, y coordinate on this chart. For example, in this particular graphic, the x-axis represents time, and the y-axis represents total number of home runs hit. Whereas the coordinate system tells us what various dimensions of our visualization may correspond to, the scale or data type will tell you where exactly your data needs to go. There are three types of scales that we could use: numeric, categorical and time series. In this example, we're using numerical data on the y axis, the total number of home runs, and we have time on the x axis, the year. Note that both of them are a linear scale, so numbers are evenly spaced out. There are other type of scales, which we'll discuss later. And the final important ingredient of an information visualization is context. So, if your audience is unfamiliar with the data, it's your job to clarify what values in your graphic represent and explain to people how they should read your chart.



So, in this particular visualization, I've provided some context by giving the chart a title. Number of NLB home runs by year. Labeling the axes, year and total home runs, and I could do a little bit more by annotating the chart potentially, so there are a variety of things that might inform the total number of home runs hit in a particular year, maybe the beginning of the live ball era, which we might put somewhere over here. The crackdown on steroid use, which happens somewhere around the year 2000, et cetera. These annotations might help people interpret this data and understand the story of, why the number of home runs hit by year is changing over time. It's important to emphasize that data is the driving force behind all four of these ingredients.

Introducing Don

My name is Don Dini. I'm a principal data scientist here at the AT&T big data group. My background is in computer science in artificial intelligence specifically multi agent systems. Before that, I did physics but then I got, I got struck over the head with the magic of computer science. And I moved into there, and into our visual intelligence. And then I taught for a while at USC, and then moving up here to the Bay area and worked for a couple of start ups. Doing a lot of work on real time system for collecting data and doing real time analysis on them. And and now I am here.

Dons Advice on Communicating Findings

So I think my experience has been that human beings are hard wired to receive things in story form. So if you can craft a narrative behind what you are doing, it makes it easy to, it makes it it makes it more compelling. And you know, it doesn't have to be you know, like a coming of age story or anything like that. It could simply be like like there was this problem, there exists this problem. Here's what people previously did, here's why it's not any good. We did this, here's what we found and like that's the end. So there's this sort of like there's this narrative arrow that goes through it. and, so that's one and number 2, I guess the other tip that I would have is to think about your audience. Who is your audience? Is your audience technically oriented, or are they not technically oriented? Is your goal basically, is your, are you trying, is it a bunch of people that you're trying to recruit to come and join you, in which case maybe you want to show them like cool demos. Or are they you know, just just like a, a room full of your hard-nosed science colleagues who are just going to brutally scrutinize whatever you present to them in which case, you may want to do something else.

Introducing Rishiraj

My name is Rishiraj Pravahan and I'm a principal data scientist at the AT&T big data center in Palo Alto. My background actually is in high energy experimental high energy physics and I was, before here, I was at CERN for about 7 years. That's where I did my PhD and post-doc for a few years. So that's sort of where I came from. And I became a data scientist almost as a natural sort of step towards learning about data and dealing with big data, as you know, produce a large amount of data. And we have to handle it and analyze it so along the way I picked up the skills to become a data scientist, and so.

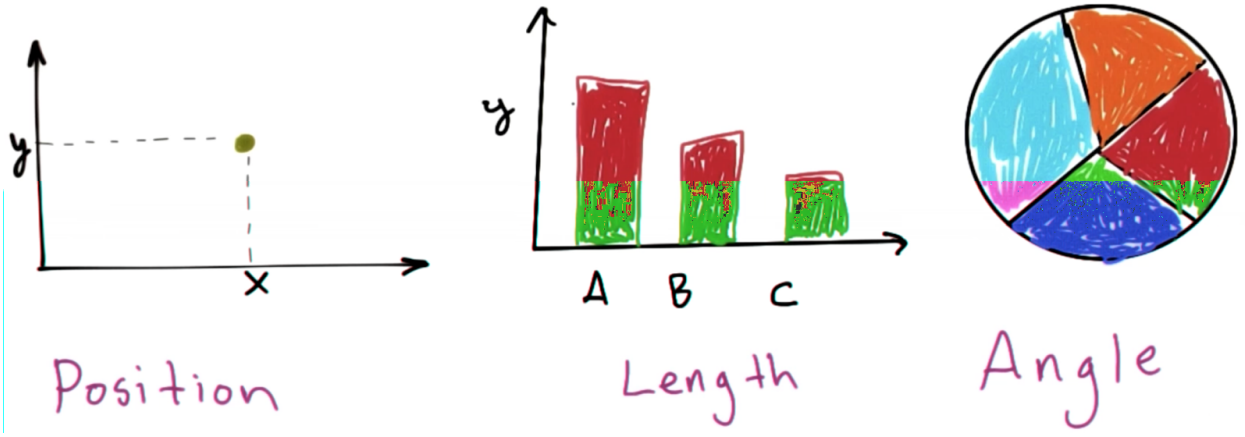
Rishi on Communicating Findings Well

In terms of communicating your findings and when you are a data scientist, you try, you tend to think in terms of code and in terms of ideas that are very mathematical etc. Or statistical you do want to have the rigor, the mathematical and statistical rigor, but you also want to make it such

that somebody who is not familiar with that, may understand, what you're talking about. So, so make it into a story. Make it into something that is easily understandable and communicable in English, for example. Or, whichever language you prefer in your group.

Visual Encodings Part 1

As we mentioned in the previous video, visual cues are one of the main ingredients to an effective visualization. In the next couple of videos, we'll cover some of the main visual cues that you can use to represent data in your charts. The first one that we're going to discuss is position. When using position as a visual cue, you compute value by looking at the position of a point compared to the position of others in the same space or coordinate system. Here, this would translate to this point's x and y coordinate. Position is useful in coding because it takes up less space than other visual cues. It's really space efficient. Each data point is simply represented as a dot. Also, points often have the same size which allows you to easily identify trends, clusters, and outliers by plotting all of your data at once. But if you have too many data points in your plot, it can be a challenge to identify what each point represents. For example, imagine that this chart were covered with dozens of green dots just like this one. It might be hard for us to identify individual dots and really understand what they mean, or what day do they correspond to. Another commonly used visual cue is length, which is often encountered in the form of bar charts. The length of a bar indicates the intensity or value of your data point. For example, the longer the bar, the greater the absolute value. One more encoding that's very common is angle.



Angle ranges from 0 to represent parts of a whole. A pie chart is a really common example of this. The sum of the wedges in this pie complete a circle and give us 360 degrees. One negative aspect to using angles as a visual encoding is that human eyes can have a difficult time differentiating angles. For example, look very similar. Because of this, angle is an encoding that you may want to avoid if you're trying to show very small differences.

Visual Encodings Part 2

Let's discuss a few more visual cues, the first of which is going to be direction. Direction is similar to angle, but the direction encoding relies on a single vector's orientation in a coordinate system. By looking at the slope of a given line, we can easily see increases, decreases, or fluctuations within our data. However, direction is tricky for many of the same reasons that angle is tricky. We can easily differentiate a horizontal or vertical line, but when we're dealing with lines with angles it's harder to see the difference between two that are both negative or positive. Another encoding that can be used is shape. Shapes and symbols are commonly used to differentiate categories of objects. For example, say that we had a scatter plot. We could use triangles and squares to show trends for two different types of data, say two different baseball teams, or two different districts in our Aadhaar data, something like that. We can also use the area or volume of our shapes to encode information.

So bigger objects typically represent greater values. Like length, when we encode information within an area or a volume, we are basically representing our data with size. So, say we were using circles. The greater the value of the data point, the bigger our circle is going to be.

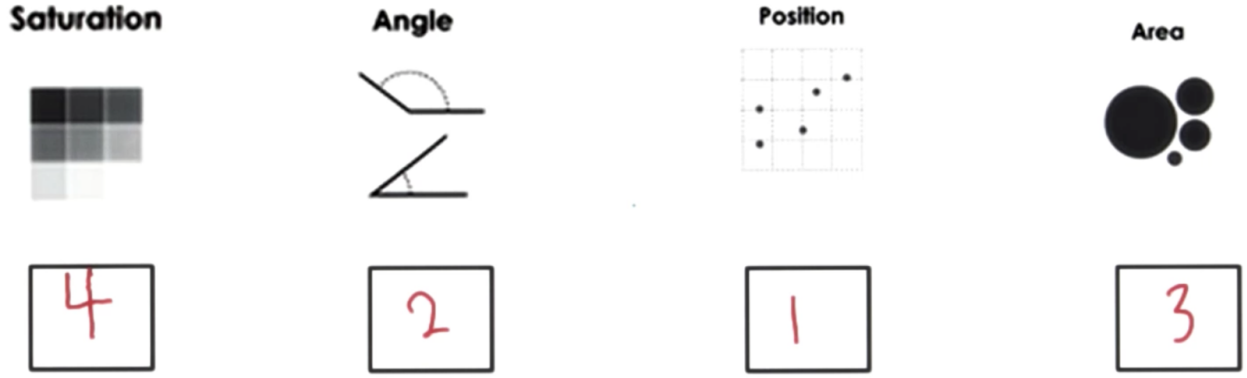
Visual Encodings Part 3

We can also encode data into our visualization using color. We can typically include values with color in 2 different ways, hue and saturation. They can be used individually or in combination. Color hue is what we typically refer to as color like red or green or blue. Different colors used together usually in the key categorical data where each color represents different group. So if we're using our baseball data maybe green is short stops, red is outfielders and orange is second basemen, something like that. Saturation is the intensity of color for a given hue. So if you select a color as blue, high saturation will be very blue and the color would look faded as you decrease your saturation. We usually use saturation to encode intensity or quantity of a value. So maybe if we were encoding the number of home runs with saturation, more home runs would be darker and fewer home runs would lighter. Or, if we were encoding Aadhaar enrollments, more enrollments would be darker and fewer enrollments would be lighter. We could also use hue and saturation in combination. So you can see here that no hue and no saturation is set to 0 and in this particular encoding, the more saturated our purple-like color is, the more negative we are. And the more saturated our green-like color is, the more positive we are. This might accentuate the differences in value and tell us when the value is positive or negative and how intense the absolute value of that effect is.

When using color to encode information, a pretty good general rule of thumb is that you shouldn't use more than a dozen colors to encode categories effectively. If you were to use more than 12 colors, it might be hard to quickly differentiate between categories, and your visualization is going to become a little bit harder to parse.

Quiz: Visual Encoding Lecture

Now that we've discussed a few visual encodings, can you rank the visual encodings listed below, with one being the most accurate and four being the least accurate? Saturation, angle, position, and area. Just type your rankings in these boxes.

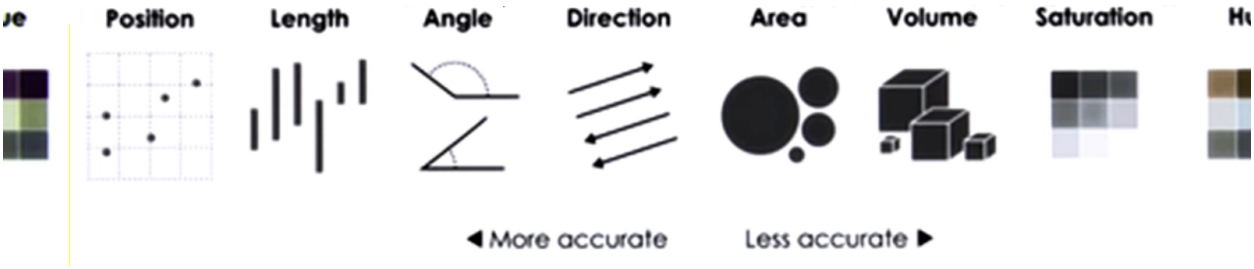


Answer:

All right, the most accurate visual encoding of these four is position. After that is angle. Area is the next most accurate. And finally, saturation is the least accurate of these four visual encodings.

Perception of Visual Cues

In 1985, two scientists from AT&T labs published a paper on graphical perception and methods. The study determined how accurately people read the visual cues that are presented here. This resulted in a ranked list of most accurate to least accurate visual cues. So you can see that position is more accurate. Where as hue is less accurate. What does accuracy mean here? Well, in this case, it just means how easily are people able to perceive the values in your data set given the visual encoding that you've chosen. You might wonder why saturation and hue are considered inaccurate. Well, this is a great example. One should be a little bit cautious when using color, hues and saturation as visual cues. Like all aspects of visual perception, we don't perceive color in an absolute manner.



For example, these are all different shades of gray. When trying to compare and contrast them, it's difficult to know which shades of gray correspond to exactly which values, or just how much darker one shade of gray is than another. For this reason, it's really hard for viewers of your visualization to really know what a different shade of gray might mean. How much bigger is one data point than another? Because of this, you should be careful when using color, hues and saturation, to encode information in your visualization. This ranking up here is really a oversimplification of how visualization works. So you should really use it as a guide, rather than a definitive rule buck. Efficiency and exactness are not always the goal of our visualization. Sometimes color, saturation or color, hue or volume or area can be really effective to communicate what we are trying to tell the viewer. However it's good to know generally how well people will be able to read different visual cues.

Plotting in Python

Now, we know a bunch about how you may encode information into your visualization and how to make an effective visualization, but we still haven't yet discussed how you can make graphics like this, short of drawing them with pen and paper. There are a number of packages for plotting in Python. One of the most popular is Matplotlib. For this course, however, I'd like to go over plotting using a Python library called ggplot, which very closely recreates the syntax used in R's ggplot2 library. If Matplotlib is so widely used, why should we use ggplot? Well, I'd like to use this package for a few reasons. First, what it produces is a bit more aesthetically pleasing than Matplotlib. Second, it's an implementation of a pretty neat concept called the grammar of graphics, which basically claims that there's a grammar involved in composing graphical components of statistical graphics. The gg in ggplot actually comes from grammar of graphics. It also plays nicely with the pandas DataFrames we've been using in this course.

To quickly summarize the ideas behind the grammar of graphics, plots convey information through their aesthetics such as x-position or y-position. The elements in a given plot are

geometric shapes, such as points, lines, or bars. Some of these shapes can have aesthetics of their own, such as their size or their color. You can think of creating plots in ggplot through the grammar of graphics as adding layers to our plot. The first step in creating a graphic is always to create our plot, which is essentially going to be our canvas. This can be done by calling ggplot data aes(xvar, yvar). Data here is going to be a pandas DataFrame, and xvar and yvar are going to be columns in that data frame. So what we're doing here is saying let's make a ggplot. The data source is going to be our data frame, and the quantities that we're interested in plotting are xvar and yvar. This might be district and number of Aadhaar enrollments or position and number of players, something like that. So what we've done here is we've made our ggplot. We've said that the data source that it will use is pandas DataFrame, and that the variables that we'll look at are xvar and yvar. This might be district and number of Aadhaar enrolled if we're using our Aadhaar data or team and total number of players if we were using our baseball data, something like that. Okay, so, so far that we've said that we'll have a plot which is mapping xvar to the x-axis, yvar to the y-axis, but we haven't said yet what type of geometric object is going to represent this data.

```
ggplot(data, aes(xvar, yvar))
+ geom_point(color='coral') + geom_line(color='coral')
```

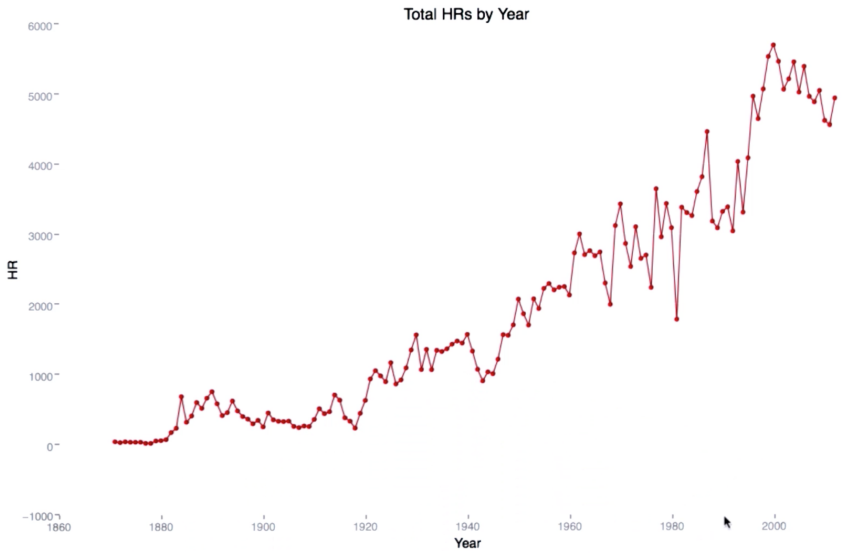
So if we add plus geom point to this statement, we'll create a scatter plot. If we also add plus geom line to the graphic, we'll connect all these points to each other with lines. Now, say that we wanted these points to have a particular color. We can pass color equals coral into geom point, and also pass color equals coral into geom line. And after we do that, both the points and the lines will have the color coral. This is the second step of making a plot in ggplot, that is choosing which type geometric objects will represent the data. The final step here is going to be adding some labels so that our plot will have some context, like a title or an x-label or a y-label. This can be done much in the same way that we added the points and lines to our plot. We can add a ggtitle to title our plot. An xlab, which will be x-label, to provide an x-label. And a ylab to do the same with the y-axis. Now all I have to do is precede this entire command by Print. And I'll produce a plot in Python. Why don't you try implementing these ideas to create a graphic of your own?

Programming Quiz: Plotting in Python

Lets assume that we have a csv file called hr_year.csv, which contains two columns, yearID and HR. YearID is the year and HR is the total number of home runs. With these two columns, the csv file contains the number of home runs hit in Major League Baseball every year. Can you write a function line plot that creates a pandas dataframe from this csv file and then creates a chart with points connected by lines, both colored red, showing the number of home runs by year? You want to do this using the GG plot library with the syntax that we just discussed. Your code should go here.

Answer:

All right. Let's take a look at the code here. So first we create a pandas dataframe called hr_year, that reads in the hr_year.csv file. Then we just print ggplot where we say that our data source is going to be the hr_year data frame. And the variables that we want to plot are year ID and home runs. Then we add geom points, with the color red. Then we add geom lines to the color red and then let's add some labels so our plot is able to interpret. So we'll title it total home runs by year. With the x label year and the y label home runs. We just print this and we'll generate our chart. Now let's see what this produces. Here's the plot that this generates. We see that we have a bunch of points that are red, where we indicate the number of home runs hit every year. They're connected by red lines, which gives us some sense of the rate of change. We have years on the X axis. Home runs on the Y axis. We have the labels that we put in. And also the title that we told our plot to show.

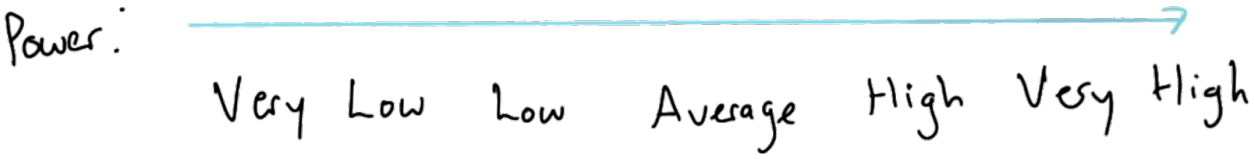


Different Data Types 1 - Numeric Data

So we've talked about visual cues to encode and represent your data. Another thing we consider is the different types of data that are available. Most data can be categorized into 3 basic types. In fact, we've seen all 3 types of this data during our walk through of the baseball data without explicitly referring to them as such. The first of these types is numeric data. Numeric data as you might expect, is any data where our data points are exact numbers. These data have meaning as a measurement such as a baseball player's height or weight or as a count, such as number of hits or home runs for a player or how many players there are on a team. Statisticians also might call numerical data, quantitative data. Numerical data can be characterized into discrete or continuous data. Discrete data has distinct values whereas continuous data can assume any value within a range. For example, a player's number of home runs would be a discrete data set. You can only have discrete whole number values like 10, 25, or 34. A player cannot for example, hit 14.375 home runs. A player either hits a home run or he doesn't. On the other hand, continuous data are numbers that can fall anywhere within a range. Like a player's batting average which falls between 0 and 1000. So a player could have a batting average of 0.250, they could also have a batting average of 0.357 or of numeric data might be a baseball player's height or weight or their number of home runs or a number of hits or a number of doubles. The takeaway here is that this is data that are numbers and they are not ordered in time. They're just numbers that we've collected.

Different Data Types 2 - Categorical Data

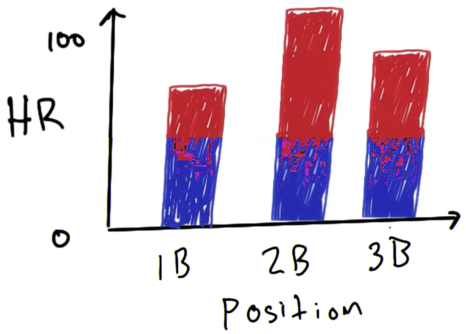
Categorical data represents characteristics, such as a player's position, team, hometown or even, handedness. Categorical data can take on numerical values. For example, maybe we'd use 1 for first baseman and don't have a mathematical meaning. That is, we can't add them together or take the average. There's also something called ordinal data, which in some sense is a mix of numerical and categorical data. In ordinal data, the data still falls into categories, but those categories have some order or ranking. For example, a scout might rank a baseball player's power on a scale from very low to very high. Another example is that you might give a movie anywhere between 1 star and 5 stars and these might be ordinal categories. For plotting purposes, ordinal data is treated much in the same way as categorical data. But the groups are usually ordered from lowest to highest, so that we can preserve this ordering.



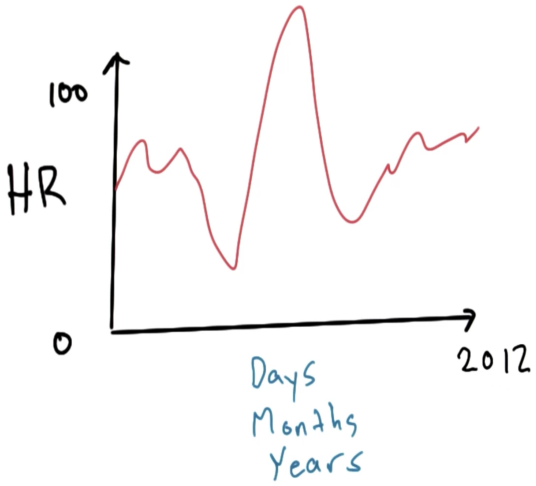
Different Data Types 3 - Time Series Data

The final type of data we'll talk about is time series data. Time series data is simply a collection of observations obtained through repeated measurements over time. In plain English, a time series is simply a sequence of numbers collected at regular intervals over some period of time. For example, we might measure the average number of home runs per players for many different years. Time series data is not so different from numerical data. The real difference here is that rather than having a bunch of numerical values which don't have any time ordering, time series data does have some implied ordering. There's a first data point collected and a last data point collected.

Data Scales



Now, let's talk about scales for categorical data. A categorical scale provides visual separation for different groups. And it often works in tandem with a numerical scale, as we'll see here when we plot home runs on the y-axis. A bar plot, for example, can use a categorical scale on the x-axis, and a numerical scale on the y-axis. So let's say that we were measuring home runs by position. The positions we've used here are unordered, but you can imagine that if we have ordinal data, we'd want these categories represented in the order that the ordinal data suggests. That will make it easier to compare and contrast your bars, or whatever encoding that you've chosen to represent your data. Finally, let's talk about time series data. The thing is, we can measure time with various different granularities. So we could use months or days or years on this x-axis, which will let us visualize our data in a discreet way. Depending on what you're trying to communicate with your data, you might use a different time scale here. So, if you're looking at fluctuations that happen on a very short time scale, days are more appropriate, but if we're looking at long term trends, we may want to look at data on a month to month or year to year basis.



Quiz: Improper Use of Scales

Alright, so we've talked about how proper scales can help present data more effectively, but why don't we look at an example where scales are not used well? When used incorrectly, scales can misguide or confuse readers. This visualization from New South Wales shows that the NSW health system is recruiting more nurses. But, I can tell you it's not a good visualization. Can you tell why? Enter your answer in the text box below.

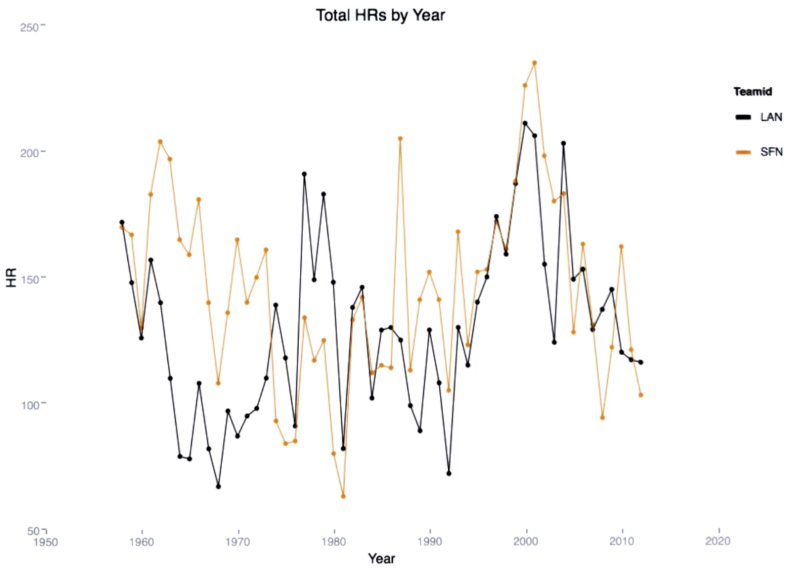
Answer:

The problem here is that the scale used here is inconsistent. Four stick people are originally used to represent 43,000 nurses. However, 28 stick people are then used to represent an increase of just eight stick people are used to represent an increase of only about 1,000 nurses. As you can see, this growth doesn't follow a linear or logarithmic scale. It's just kind of random, and it's very confusing to the reader. You'd look at this and think that the number of nurses exploded, but it really only increased by 3,000. We can only assume that some kind of mistake was made here, that this doesn't efficiently communicate what's actually happening.

Quiz: Plotting Line Chart

Now that we've discussed it from different visual cues and different data types, as well as the scales to use for those data types, why don't we circle back to making graphs with ggplot and try to do something a bit more advanced. This time, we'll want to write a function lineplot_compare, that will read in a CSV file called hr_by_team_year_sf_la.csv. This file contains 3 columns - yearID, HR, and teamID, which represent the total number of home runs hit each year by either the San Francisco Giants or the LA Dodgers. Why don't you produce a visualization comparing the total home runs hit by year for the 2 teams. Note that to differentiate between multiple categories on the same plot in ggplot, we can pass color in with the other arguments to aes rather than in our geometry functions. For example we could say ggplot data aes xvar, yvar, and then color=catetory_var This should help you to make this chart. Your code goes here.

Answer:

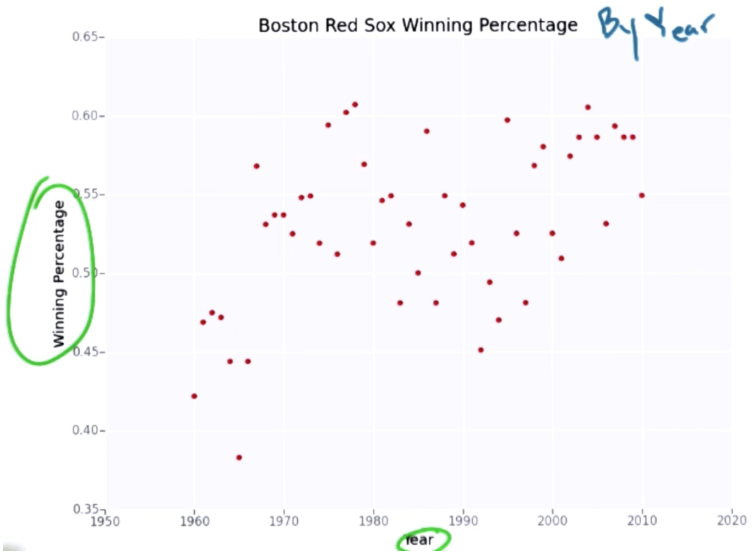


So first we're going to make a pandas dataframe. Again, I'm going to call it hr year. That will read in hr by team year sf ls.csv. Then we'll create our ggplot. So we'll say print ggplot hr year, and note that we say aes yearID, HR, and then we set color equal to teamID. Then we're going to add geometric points and geometric lines, but not pass on a color. Otherwise this is very similar to our basic plot, which didn't have different lines and points for different teams. Let's see what this creates when we actually run it. As you can see we have here a chart which compares the total number of home runs hit per year, for the L.A. Dodgers, called here LAN for LA National League. And the San Francisco Giants, called here SFN for San Francisco National League. We have two different sets of points and lines which are color coded according to team, and we're able to compare how the total number of home runs hit have varied by year for the two teams.

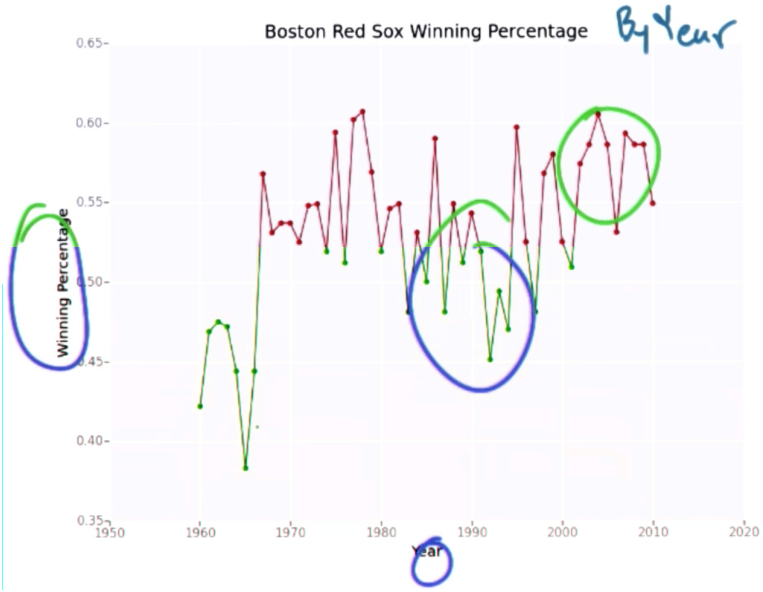
Visualizing Time Series Data

All right. Now that we've learned the basics of visual cues, different data types, and different skills that we might use to plot those data types. Why don't we focus in more on time series data? Since that's the format that our New York City subway data and weather data will come in. We'll discuss different things we can include in the plot for time series data. Starting with something simple like a scatterplot, and adding more complex features like maybe a LOESS curve. With each iteration, we'll show how we can combine visual cues, data types, and appropriate scales in our charts to achieve visual excellence. Again, why don't we use baseball data as an example. Let's plot the winning percentage of my least favorite team as a Yankees fan, the Boston Red Sox, from 1960 until 2010. Let's start with a simple scatter plot.

Scatter Plot



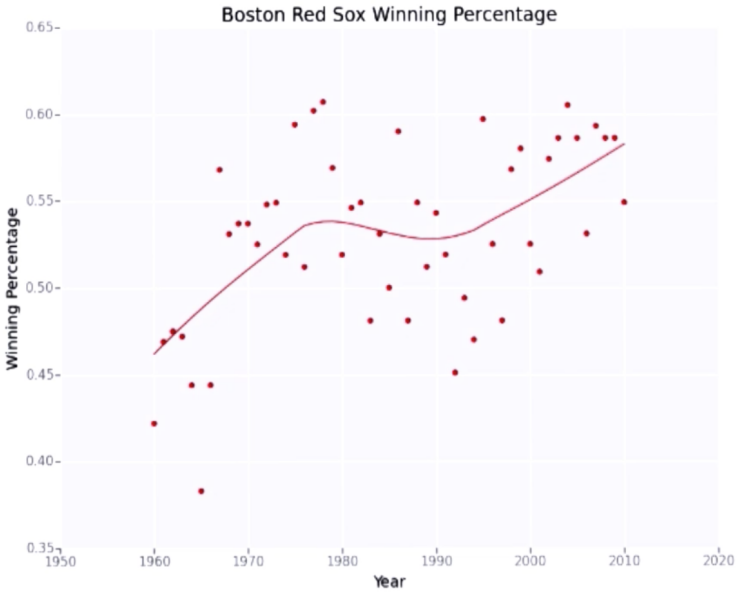
Let's start with a simple scatter plot. Each dot in this chart represents the winning percentage of the Boston Red Sox in the year between 1960 and 2010. In this chart, the main visual cue that we're using is position. The year for our particular winning percentage is dictated by where we are in the x coordinate. And we see what the winning percentage is by looking at a data point's position on the y-axis. Secondly, we can note that this plot provides pretty good context. We have an X labeled Year, indicating what the x-axis represents. A Y labelled Winning Percentage, telling us what the y-axis represents. And we have a pretty descriptive title. Boston Red Sox Winning Percentage. We might include by year to make it even better. This will help people who are unfamiliar with this data really look at this chart and understand what's happening. But as you can see, a scatter plot here really puts focus on the individual values. It's a little bit hard to discern trends. It could also be hard for a reader to mentally fill in the gaps between the points and really understand what's happening from year to year. It's for this reason that we might want to use a line chart.



Line Chart

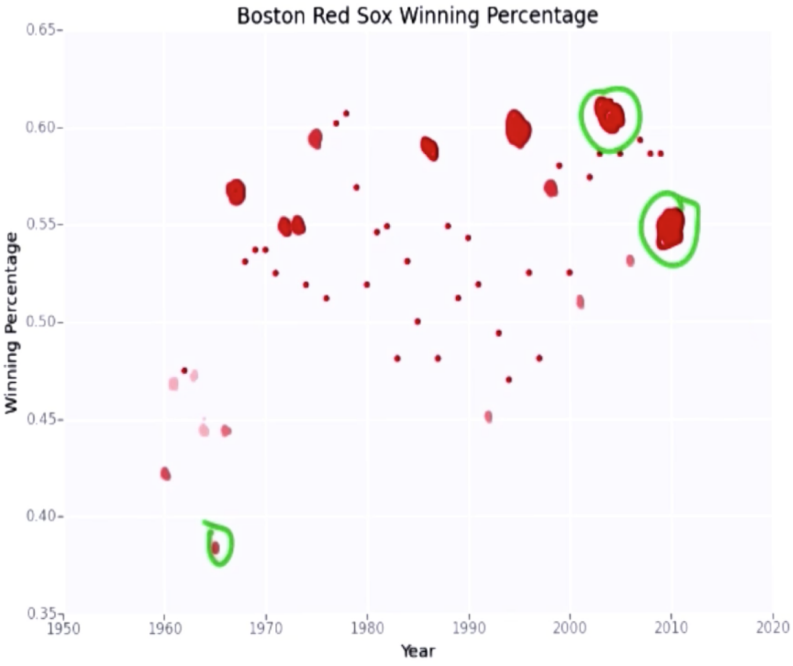
So, to mitigate the shortcomings of a scatter plot, we can plot the same data as a line chart, and connect the dots with lines, as we see in this image. Putting lines here really emphasizes trends. So, as we can see, there was a pretty bad series of Red Sox teams in the 90s. And that the Red Sox teams have been doing pretty well in the 2000s. It's a little bit harder to see this when we just have the points without the lines. That being said, having lines that connect each point really puts a focus on year to year variability. We can see, for example that from 1964 to bit harder to have a good sense of the more global trends. Are the teams getting better or worse on average? For this reason, we might introduce a lowest curve instead of these lines.

LOESS Chart



If we wanted to emphasize long term trends in our data, rather than year to year variability we might plot a LOESS curve over our data points. LOESS is a form of weighted regression. We won't go into it too much in this course, but you can see here that it captures the overall trends in our Red Sox data rather than the year to year effects. So we see that the themes were getting better up until the 80s, got a bit worse in the 90s, and have continued to improve in the 2000s. Adding this line makes it easier to take a quick look at our chart and come away with an understanding of the big picture. Where as if we just had the points or have the lines connecting each point, it requires a little more heavy lifting by the viewer.

Multivariate Data Part 1



All right. So it's great that we can visualize the winning percentage of the Red Sox from 1960 to 2010. But what if we wanted to do a slightly more complex analysis to try to understand what is driving that winning percentage. We'd have to incorporate more variables. But how? One way is that we can use the visual cues that we're familiar with to encode even more information into the chart. For example, let's go back to our scatter chart. We can use the size of the points to represent the number of home runs hit by the Red Sox players that year. Dots that are larger meant that more home runs were hit that year whereas smaller dots mean that fewer home runs were hit. Hopefully, what we would see is that bigger dots occur when the winning percentage is higher. I've filled the plot in with some dummy data just so that you have an idea of what this might look like. We can even double up on visual cues. For example maybe we also use the saturation of the red color to indicate how many home runs are hit. So our larger dots will be a more intense shade of red, whereas our smaller dots will be a less intense maybe almost pink shade of red. A redundant visual cue can help reinforce what might be a challenge

to see if we only used one visual cue. So by using both size and color here, hopefully we really hammer home to the viewer that there were more or fewer home runs in a given year.

Multivariate Data Part 2

So as you can see, applying the right types of encodings and additional details to our charts whether it be lines or, you know, additional colors or sizing on our points we can really transform our visualization from just an uninspiring dot chart to a better chart that's easier to read. Where the underlying relationships between various aspects of our data Can be more easily comprehended and visualized by our viewers.

Rishrajs Advice to You

My tips for aspiring data scientists would be to sort of learn the tools for data science very well, very, very well. And of course through several curriculum, for example, Udacity from university courses, etc. Moreover, to use them in the correct way. You know, one of the analogies I like using is Galileo, before Galileo, the telescope existed, but he used it to look at the stars and it changed the course of history. So it's important to know your tools, how to use them, and use them to do something innovative and new and different. And that's my advice.

Dons Advice to You

The success or failure of the models that you create is in a very large part determined by essentially what are you paying attention to about your data. So, this is the problem called feature selection. And I think that not enough time is spent in doing this. It's very, very important so it's, in a lot of data science shops, it's very easy to just sort of take, records that come in, and then put them into a logistic regression, and you get some output, and then like, you can call it a day. and, you know, maybe often times that could be enough to take you, like, you know, 50% of the way there. But really the difference between, like, a mediocre and a top-notch shop, is like you have sat and thought about what are the things to pay attention to? Like should I synthesize these features into a new feature? Is there something new about this phenomenon that I should be capturing but haven't yet captured? So there's a lot of sort of there's a lot of process that goes into, just the feature selection process which is, is hugely important. More advice, which just occurred to me, is there's an additional, there's a world of tools out there that are not merely limited. so, so everything you would learn in the machine learning class is fantastic. It forms a great foundation. Beyond there, there's a wealth of mathematical tools, and the more of them that you know, the more that it will help you. So things, for example, like differential equations, or stochastic processes such as Markov chains. Knowledge of these things allow you to break down a problem that you know nothing about. You can sort of break it down a little bit, a little bit further into pieces that give you just that extra bit of predictive ability or or insight into a problem. And the more tools that you have at your disposal the more it will help you.

Lesson 4 Recap

All right. So why don't we recap what we've discussed in this lesson. First, we discussed the components to an effective information visualization. They were visual cues, the coordinate system, the scale and data types, and the context. After that, we dived a little bit deeper into the various different ways that we can encode data visually. These included its size, position, color, and many more. We also discussed various scales and data types such as numerical, categorical, and time-series. We took a deeper dive into time-series, since that's what most of the data for our assignment will look like. And finally, we went over the basics of creating graphics using ggplot, a library for Python. For this week's assignment, you'll use what we've learned in this lesson. That is the basic principles of data visualization and the basics of how to create a graphic in ggplot. You'll create a few basic visualizations that describe things about the New York City subway. So, you'll want to communicate to people our statistical findings from assignment 3, but also maybe just some general information regarding the subway. Ridership at various hours of the day, how many people are going into and out of different stations, etc.

Lesson 4 Conclusion

In this lesson, we've discussed some of the basic skills necessary to make an effective information visualization. We've talked about some of the best ways to visually encode data, and also some of the most effective methods for charting different types of data. We've also gone over the basics of making visualizations in Python. If you find the individualization really interesting, I'd strongly encourage you to look at some blogs or websites that aggregate some of the most interesting data visualizations. It's not hard to find them. Just go to Google, type in data visualization, and you'll come across some really interesting stuff. You'll use this new found data visualization know how to make some really interesting graphics that communicate our findings so far about subway ridership in New York City. I'll be really interested to see what you guys come up with.