Visual Encoding

Michael Dubakov, TargetProcess Founder
September, 2012

How do people perceive information? How designers can thrive on this process to help people understand data faster? Let's try to look into this complex field and explore some basic principles.

> *The visual encoding is the way in which data is mapped into visual structures, upon which we build the images on a screen.*

There are two types of visual encoding variables: planar and retinal. Humans are sensitive to the retinal variables. They easily differentiate between various colors, shapes, sizes and other properties. Retinal variables were introduced by Bertin (→) about 40 years ago, and this concept has become quite popular recently. While there's some critique about the effectiveness of retinal variables (→), most specialists find them useful.

The goal of this article is to provide an engaging introduction to visual encoding, and to give some hands-on examples of how it helps to present data in a meaningful way.

Data types

We'll start with some complex things: data types (→). There are three basic types of data: something you can count, something you can order and something you can just differentiate. As often is the case, these types get down to three un-intuitive terms:

Quantitative

Anything that has exact numbers.

For example, Effort in points: 0, 1, 2, 3, 5, 8, 13.

Duration in days: 1, 4, 666.

Ordered / Qualitative

Anything that can be compared and ordered.

User Story Priority: Must Have, Great, Good, Not Sure.
Bug Severity: Blocking, Average, Who Cares.

Categorical

Everything else.

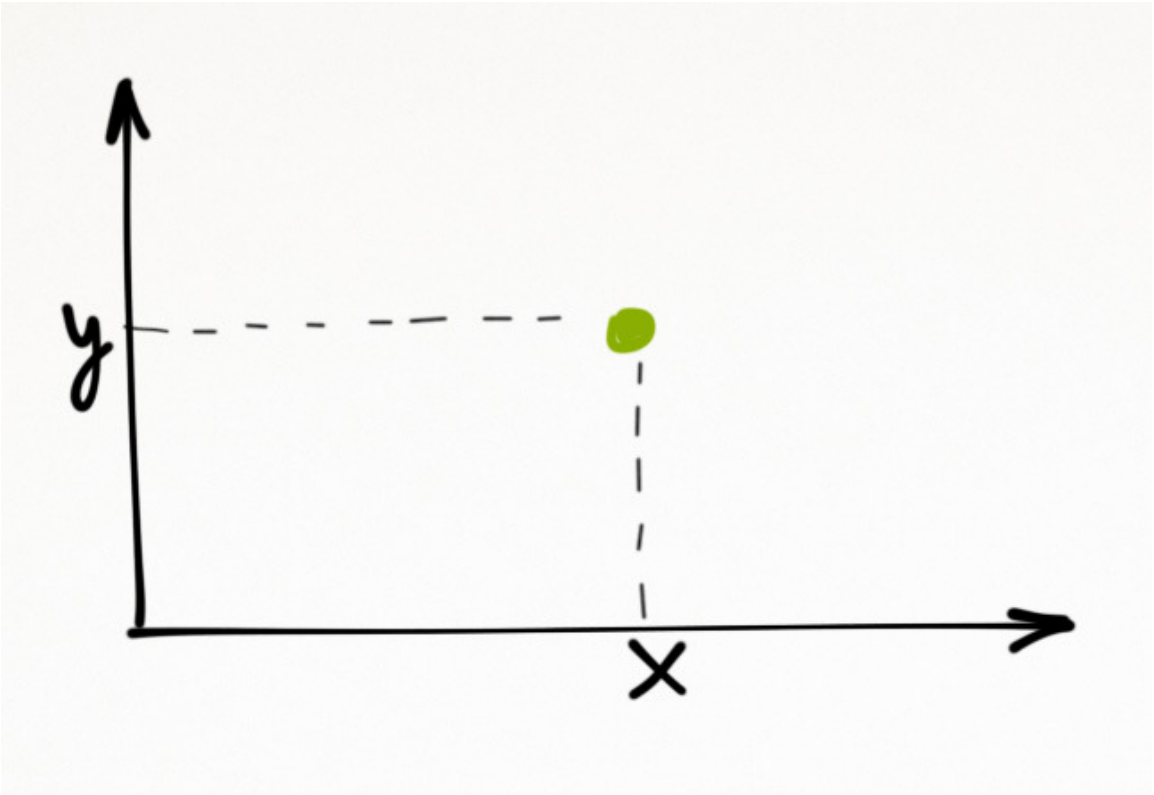
Entity types: Bugs, Stories, Features, Test Cases.
Fruits: Apples, Oranges, Plums.

Planar and Retinal Variables

OK, we've got some data. Now how do we present it? We have several visual encoding variables.

X and Y

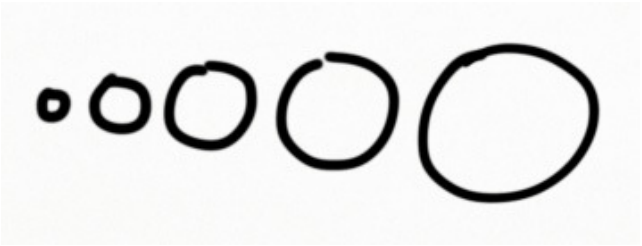
Planar variables are known to everybody. If you've studied maths (which I'm sure you'd have), you've beed drawing graphs across the X- and Y-axis. Planar variables work for any data type. They work great to present any quantitative data. It’s a pity that we have to deal wuth the flat screens and just two planar variables. Well, we can try to use Z-axis, but 3D charts look horrible (→) on screen in 95.8% of cases.



So what should we do then to present three or more variables? We can use the retinal variables!

Size

We know that size does matter. You can see the difference right away. Small is innocuous, large is dangerous perhaps. Size is a good visualizer for the quantitative data.



Texture

Texture is less common. You can't touch it on screen, and it's usually less catchy than color. So, in theory texture can be used for soft encoding, but in practice it's better to pass on it.

Shape

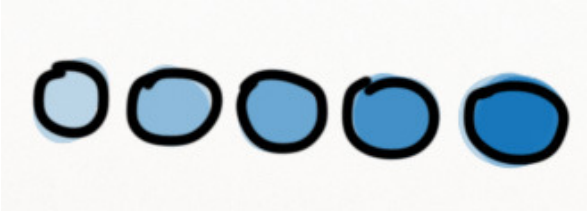
Round circles ○, edgy stars ☆, solid rectangles ▉. We can easily distinguish dozens of shapes. They do work well sometimes for the visual encoding of categories.

Orientation

Orientation is tricky.
While we're able to clearly identify vertical vs. horizontal lines, it is harder to use it properly for visual encoding.

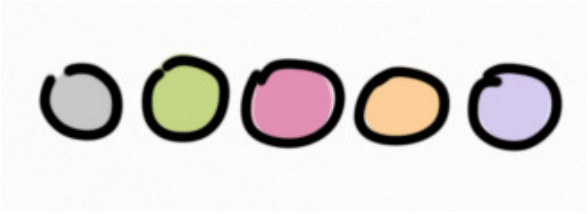
Color Value

Any color value can be moved over a scale. Greyscale is a good example. While we can't be certain that #999 color is lighter than #888, still it's a helpful technique to visualize the ordered data.



Color Hue

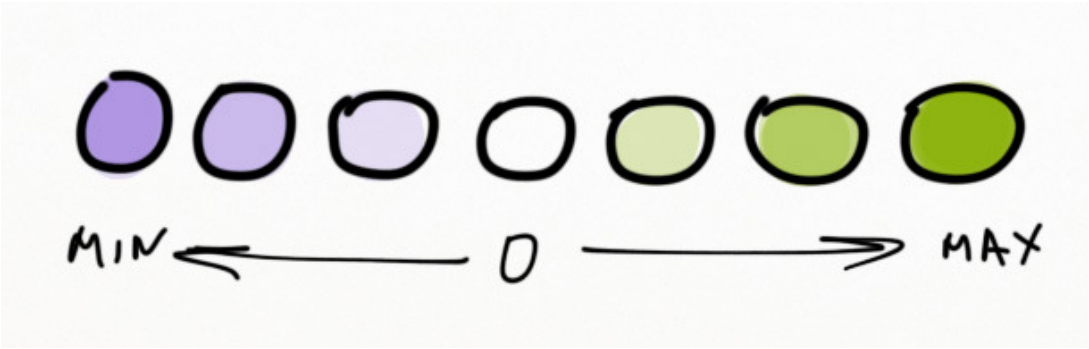
Red color is alarming. Green color is calm. Blue color is peaceful. Colors are great to separate categories.



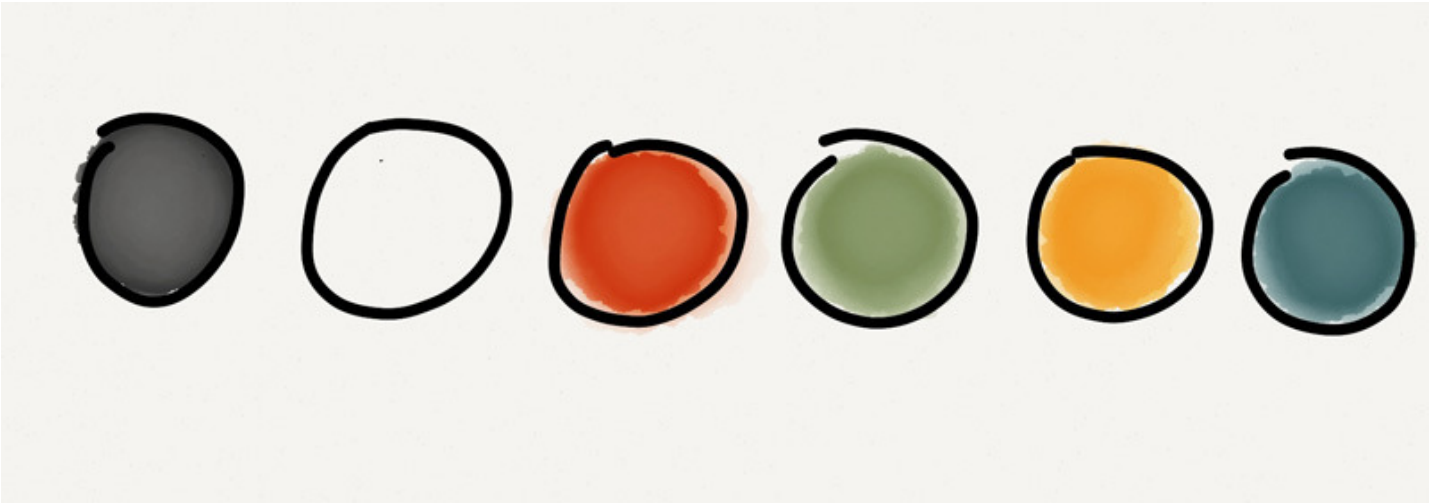
Color in More Detail

Color is the most interesting variable, let’s dig into some details here. There are three different scales that we can use with color. We've already mentioned two of them: the categorical scale (color hue) and the sequential scale (color value).

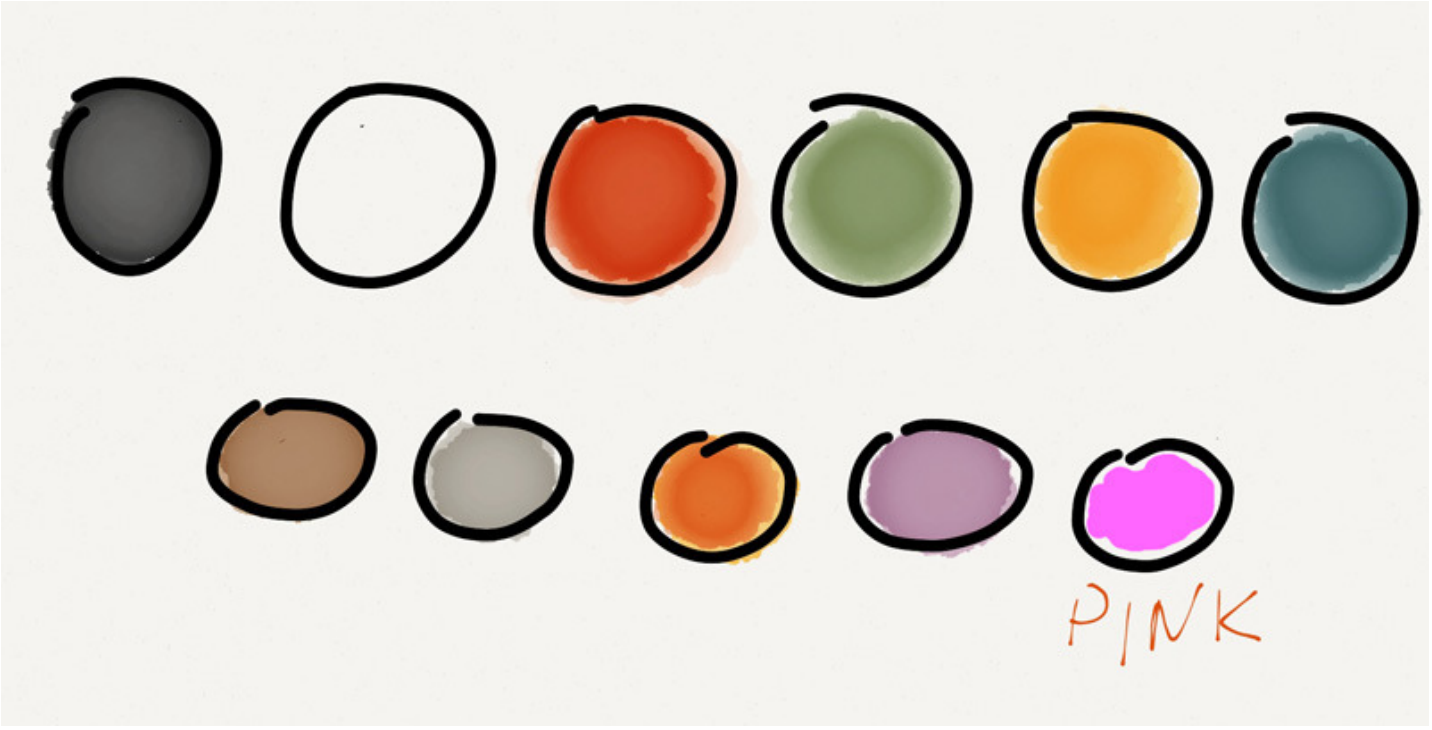
Diverging scale is somewhat new. It encodes positive and negative values, e.g. temperatures in range of -50 to +50 C. It would be a mistake to use any other color scales for that.



There are six primary colors:



The general rule of thumb is that you can use no more than a dozen colors to encode categories effectively. If there's more, it'd be hard to differentiate between categories quickly. These are the most commonly used colors:



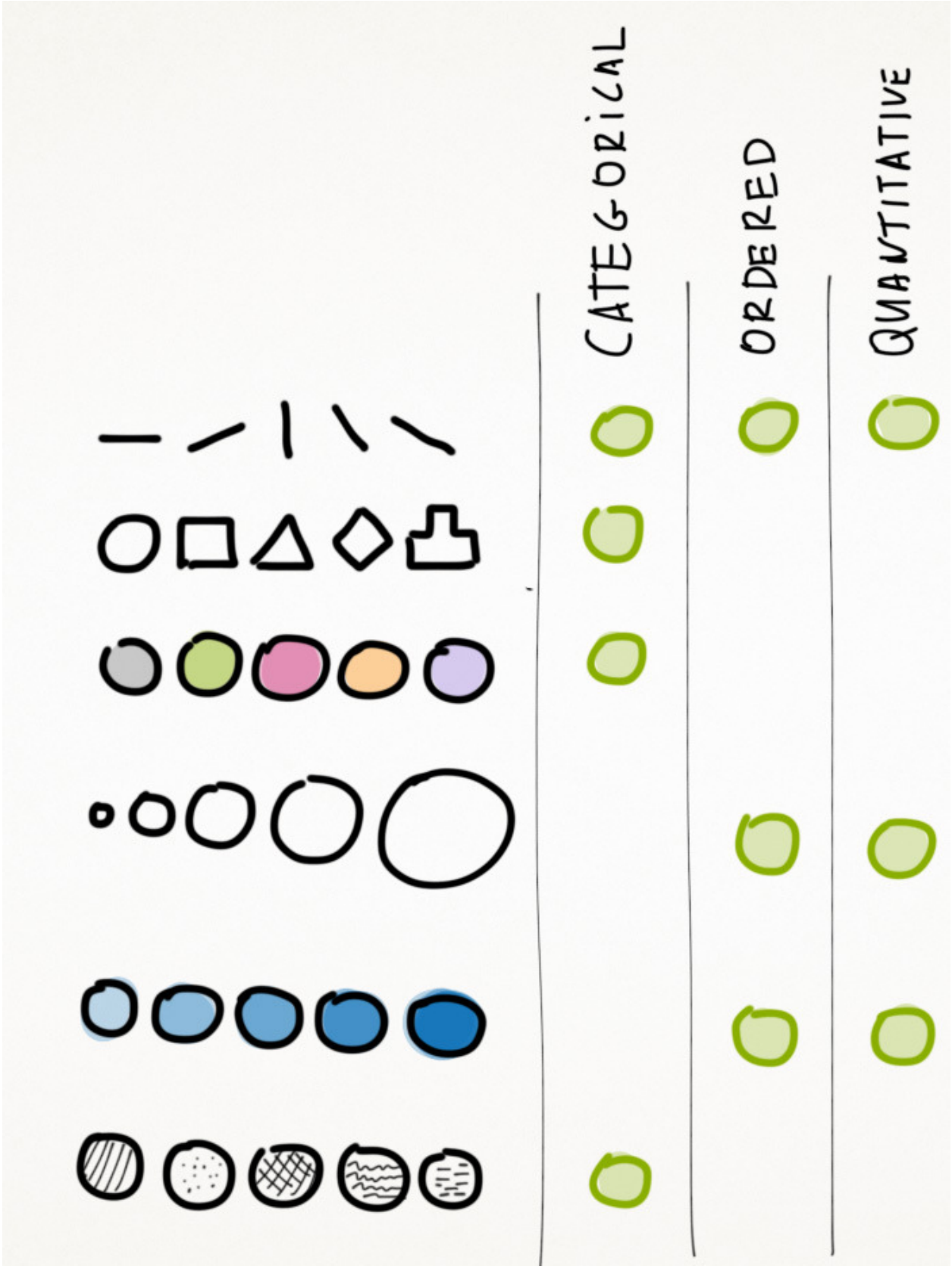
> *“Avoiding catastrophe becomes the first principle in bringing color to information: Above all, do no harm.”—Tufte*

The next obvious question is:

How to Apply the Retinal Variables to Data?

It is quite clear that we can’t use all variables to present any data types. For example, it is wrong to use color to represent numbers (1, 2, 3). And it is bad to use size to represent various currencies (€, £ , ¥). Why on Earth should small circles stand for euro, and large circles for pounds?

Here's the retinal variables usage summary:



Note that planar variables can be applied to all the data types. Indeed, we can use the X-axis for categories, ordered variables or numbers.

The Basic Example

OK, now let's tap on some techniques to visualize real data. Sample data is very simple, we just want to visualize quantity of items:

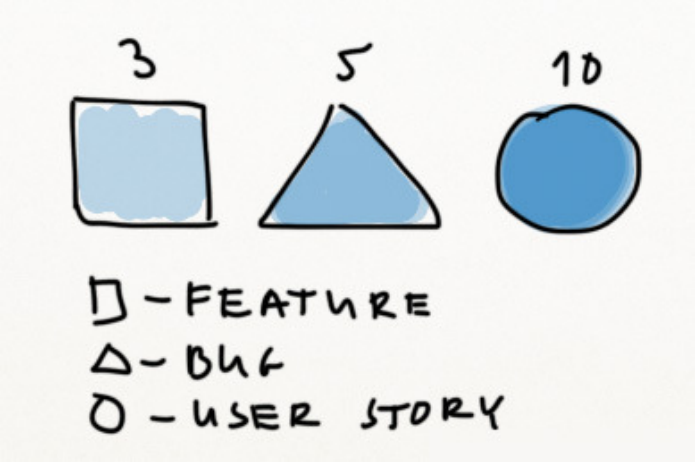
Item Type	*Quantity*
Features	3
Bugs	5
User Stories	6

We have just two variables: **Item Types** (Categorical) and **Items Quantity** (well, Quantitative). All the possible choices are based on the table above:

Item Types	Orientation Color Shape Texture X (or Y)
Item Quantity	Orientation Size Value X (or Y)

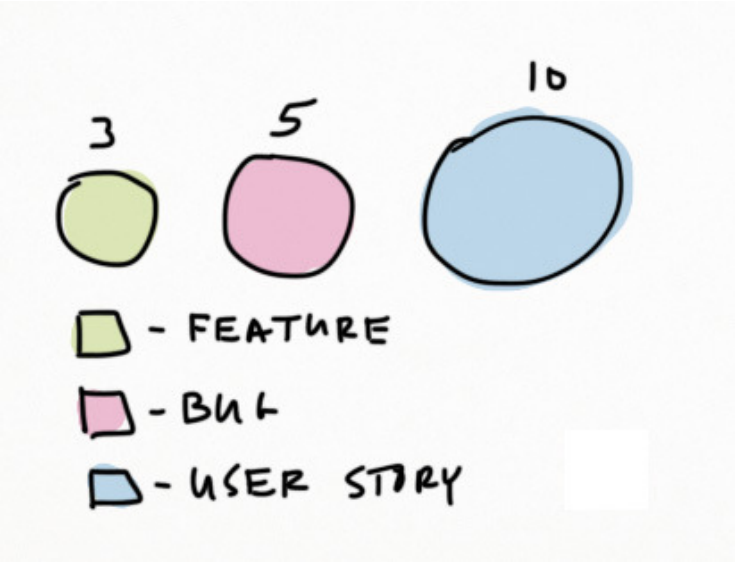
In theory, you can mix these variables as you wish. I'm going to try four combinations.

Shape + Value



Hmm, looks like a puzzle. Value doesn't work for the quantitative data, it seems. Let's try something else!

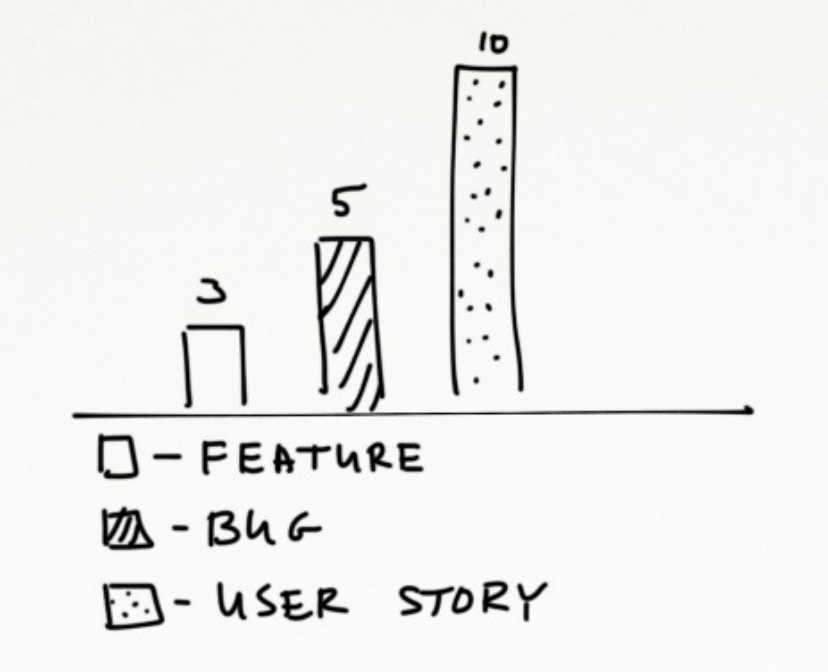
Color + Size



Well, slightly better. The color coding works for entity types. For example, in TargetProcess we've got green Features, red Bugs and blue User Stories. Still not very good.

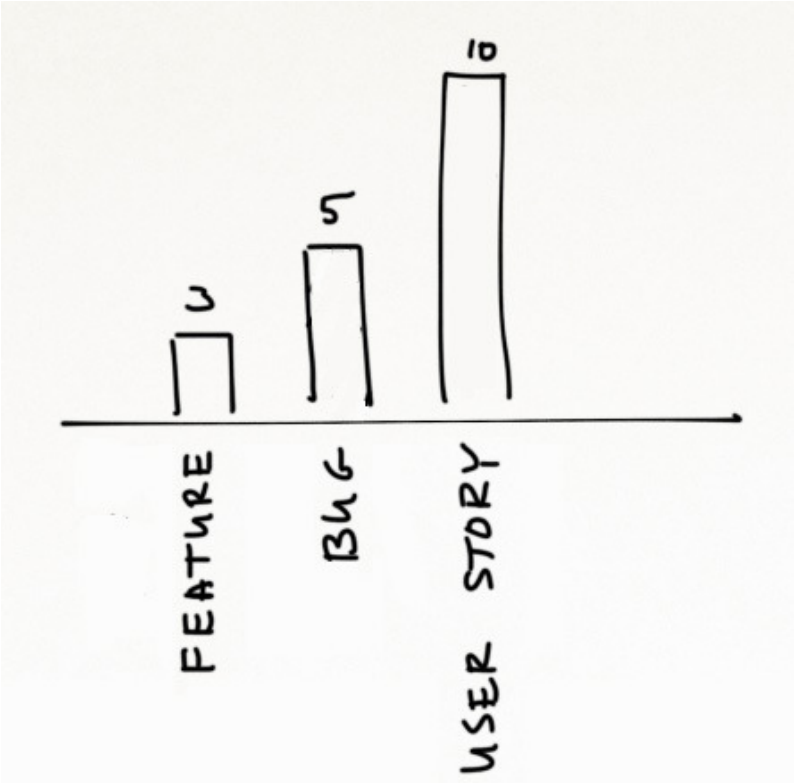
> *A very simple rule in visualizations is to never map scalar data to circle radii. Humans do better in comparing relative areas, so if you want to map data to a shape, you have to map it to it's area. (→)*

Texture + Y



Almost great. But why this legend with texture? Can we just remove it? Yes! Let's use the X and Y planar variables.

X + Y



Now we have the best result! It turned out that X+Y works great for a simple data set with just two variables. So, there's no need to use retinal variables at all.

Retinal variables should be used if you need to present *three or more* data sources.

The Four Variables Example

Three is quite trivial, so we'll take four variables. Say, we have bugs, stories, and tasks and we want to visualize some properties of these entities:

– **Types**

– **Priority**

– **Average Effort in Points**

– **Average Cycle Time in Days** (→)

Here is our data:

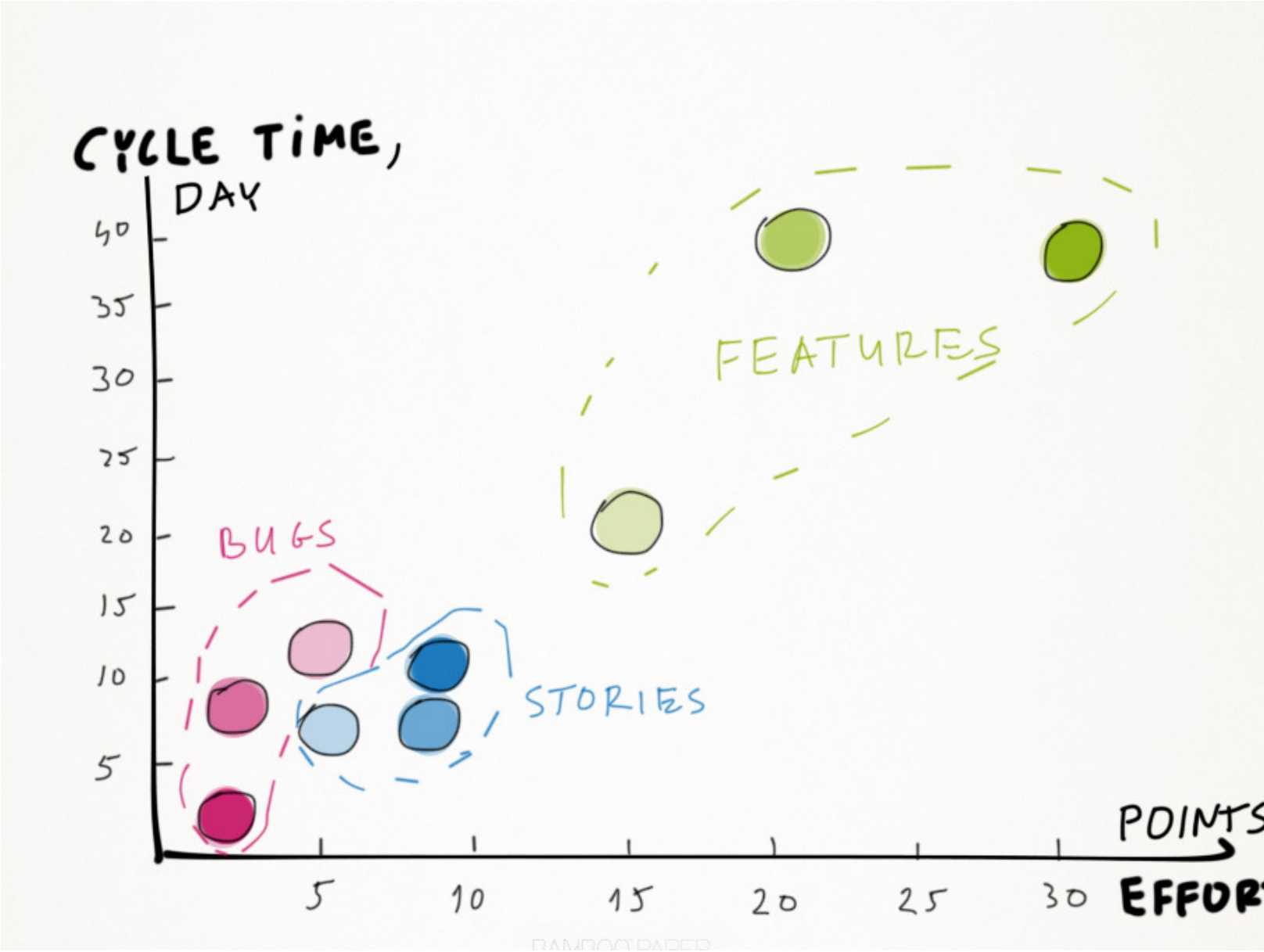
Type	*Priority*	*Average Effort*	*Average Cycle Time*
Features	Must Have	30	40
Features	Good	20	40
Features	Nice to Have	15	20
Bugs	Fix ASAP	2	2

Bugs	Fix	2	8
Bugs	Fix if Time	5	12
User Stories	Must Have	8	10
User Stories	Good	5	7
User Stories	Nice to Have	8	7

We need to pick four variables. Surely, there're other choices, but here's what I've selected:

Variable	*Type*	*Encoding*
Entity Type	Categorical	Color Hue
Priority	Ordered	Color Value
Average Effort in Points	Quantitative	X
Average Cycle Time in Days	Quantitative	Y

Now it's easy to draw the chart. The important bugs are shown in deep red, the unimportant ones — in light red. The same pattern applies to features and user stories.



What can we say about this chart? Here are some useful observations:

– Bugs are usually are smaller than user stories, and features are the largest entities.

– Important bugs are small and get fixed quickly.

– Important features are the largest, and it takes more time to release them (interesting information, by the way!).

– Unimportant bugs are the largest, and it takes longer to fix them.

– There's a good correlation between effort and cycle time: it takes more time to deliver large entities.

Of course, you can get the same info from the plain table above, but the chart is much more fun to explore.

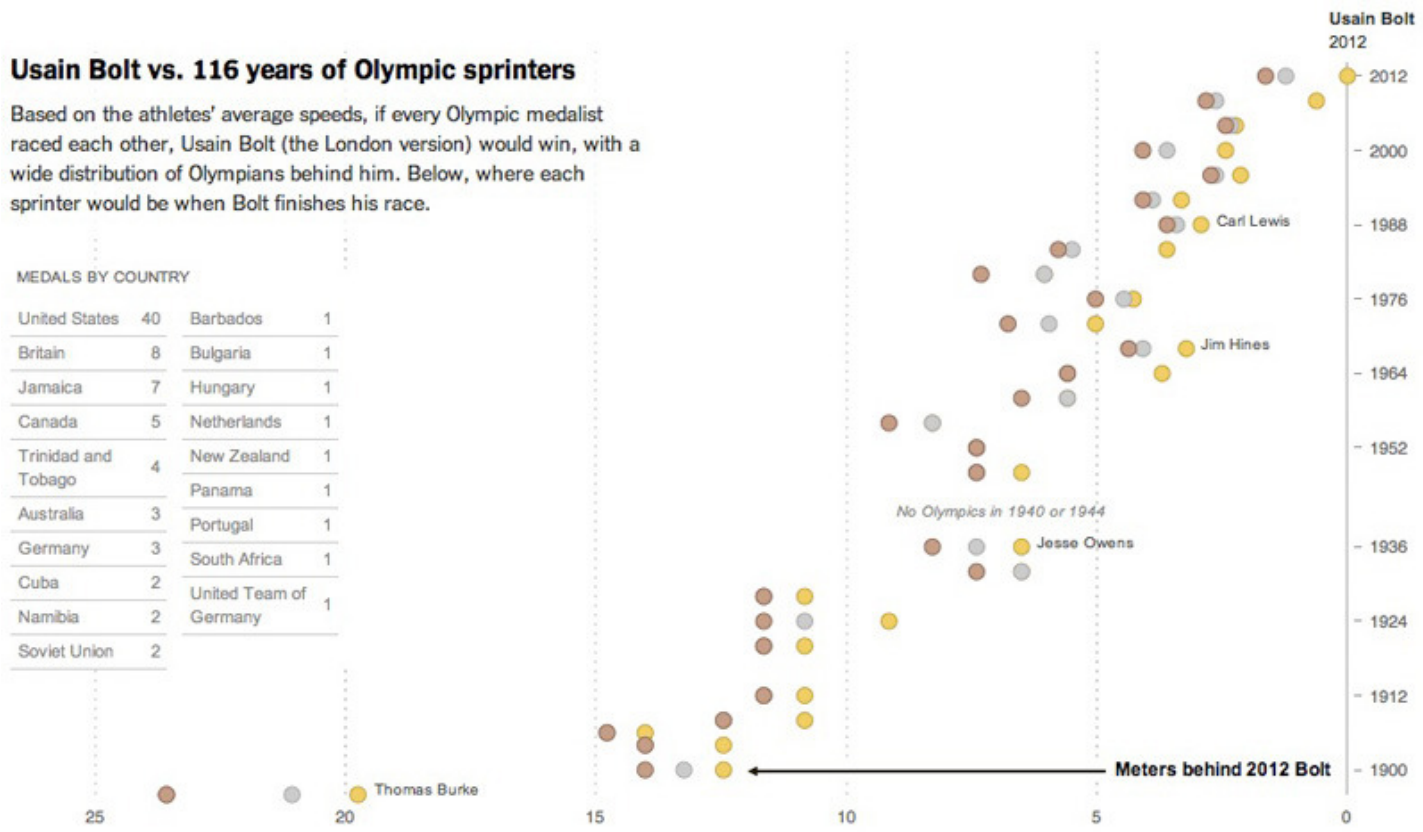
The Eight Real Examples

Let’s check some real-life examples to get an even better idea of what visual encoding is about. All

these examples are related to sports.

#1. Usain Bolt vs. The World

New York Times is a never-ending source of cool visualizations. This one is about the Olympics, the Men's 100-Meter Sprint.



Visual encoding variables:

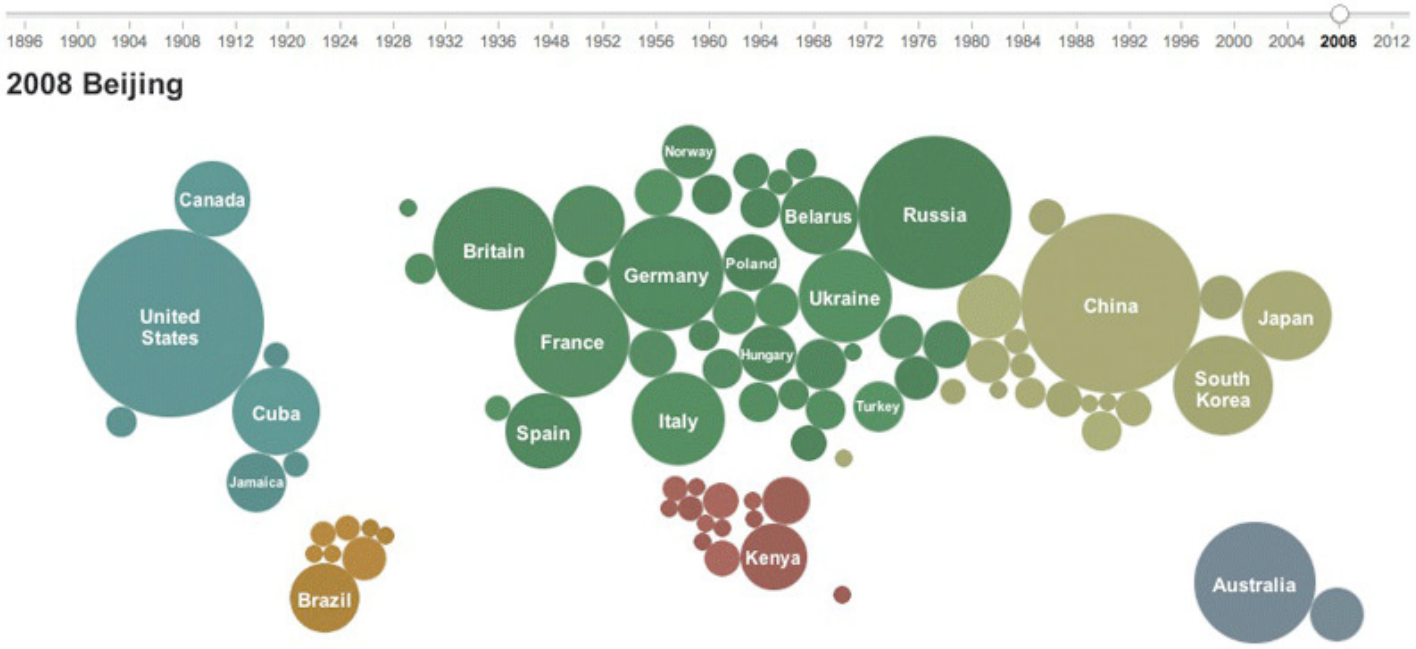
– Color: natural colors used to encode bronze, silver and gold medals

– X: meters behind Bolt (a quite unusual but very impressive metric)

– Y: year

#2. Olympic Medals

That's how the world map can be used to visualize the medal counts. Very smart. There are names for nations with many medals, all the rest can be identified by their geographic position (everybody knows where New Zealand is).



Visual encoding variables:

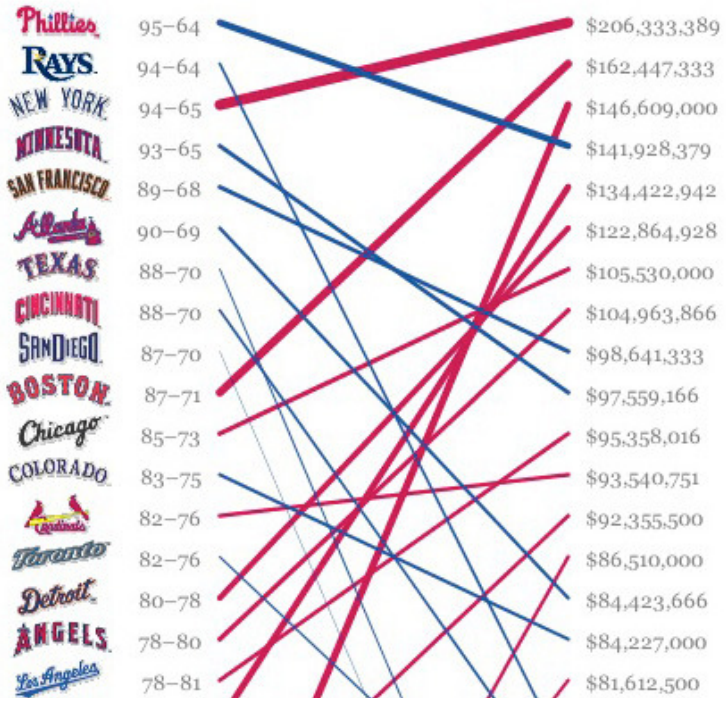
– Color: continent

– Size: medals count

– X and Y: the world map

#3. Baseball Teams Performance

The Y variable is used twice in this example. The ranking on the left shows day-to-day standings. The salaries are on the right. The lines connect teams with their salaries, the thicker the line, the higher the salary. The blue color shows that the team is doing well for its money; the red color shows the opposite.



This chart is fun to explore. We can see right away that the Rays are doing fantastic as well as San Diego and Texas, while Chicago has some problems.

It would be great to be able to focus on the blue or red teams only, this visualization lacks some interactivity.

Visual encoding variables:

– Y: baseball teams

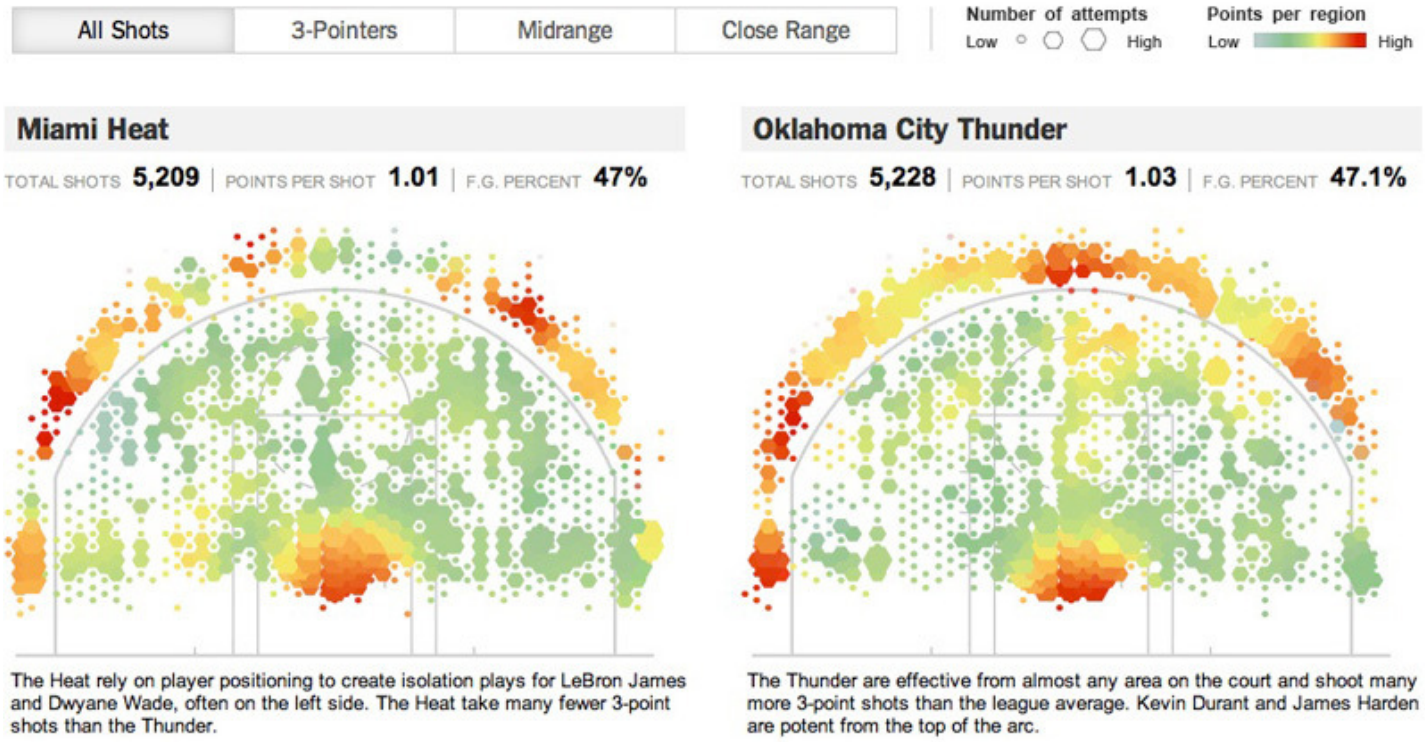
– Y: salaries

– Color: trend (good or bad)

– Size: salary

#4. Basketball Teams Performance

A heatmap is one other nice way to get the best of colors. Here's a very nice visualization of basketball teams performance created by (surprise!) New York Times. You can immediately spot hot areas on the court and compare the shot patterns for both teams. The Thunder rely on 3 pts shots heavily, while the Heat are more diverse.



Visual encoding variables:

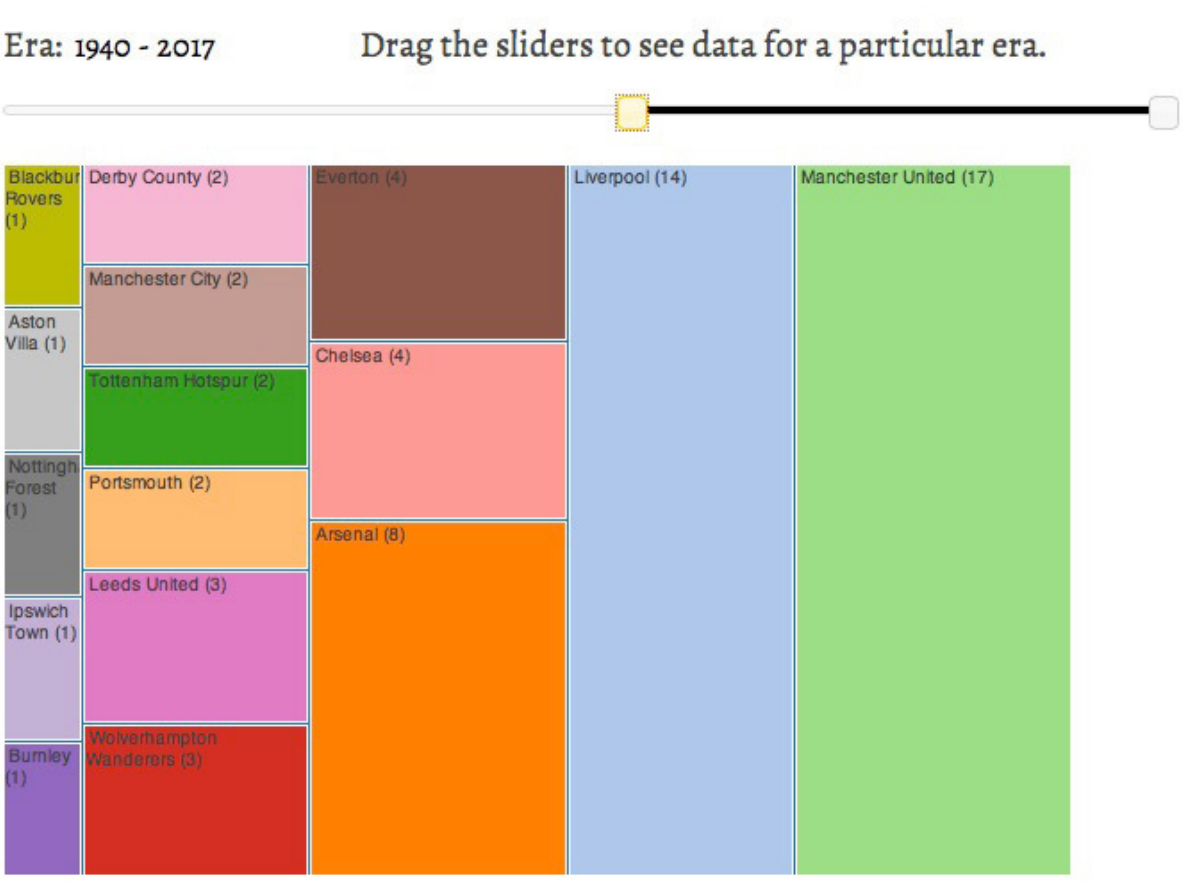
– X and Y: basketball court map

– Color: points per region

– Size: number of attempts

#5. Football Clubs League Dominance

Poor usage of encoding variables is not an exception. Most of such mistakes are related to the incorrect color choices. It might seem that color encodes a football team here, but it doesn't. When you change the range, color changes as well — this is just a random color to help you differentiate between the areas. In this case it would've been better to not use any color at all or use it wisely and encode only the prominent teams with colors.



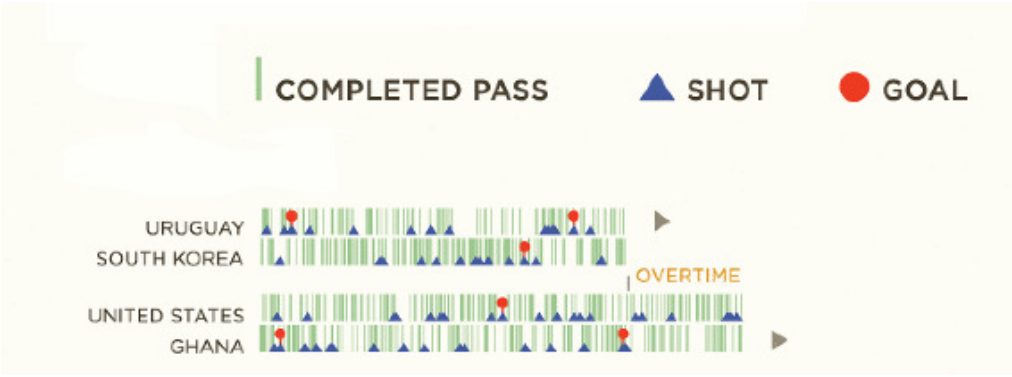
Visual encoding variables:

– Color: ???

– Size: championship years

#6. Football World Championship

This one is an offbeat visualization of South Africa's Football World Champions of 2010. Good usage of shapes and colors. The chart represents timelines of two football games.



Visual encoding variables:

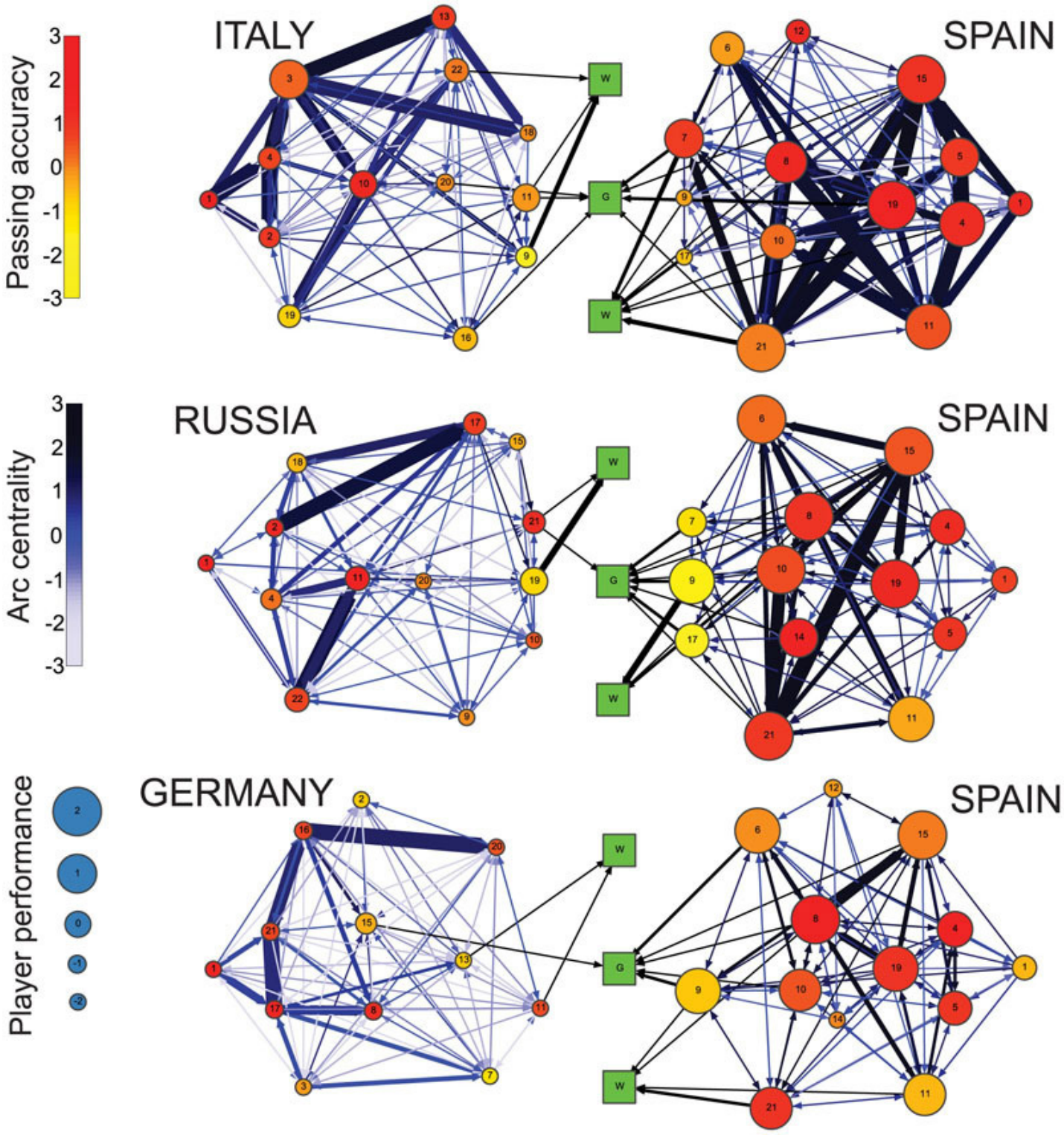
– X: time line

– Y: teams

– Shape + Color: event (goal, pass, shot)

#7. Football Teams Performance

This visualization utilizes color scale and size. However, it has a mistake. A diverging scale should have different colors for positive and negative values, but in this image we see just one color gradient.



Visual encoding variables:

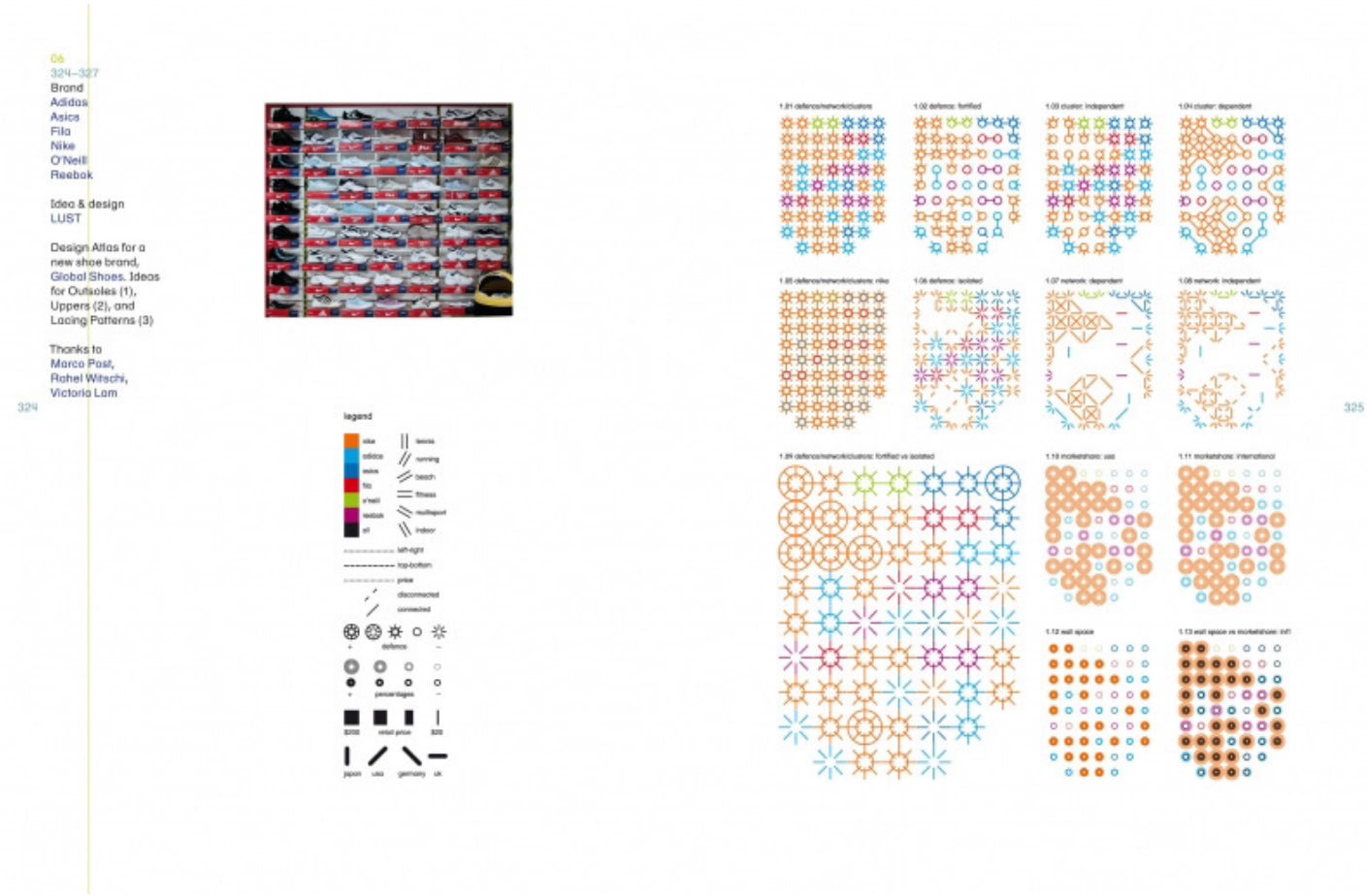
– Color: passing accuracy

– Size: payer performance

– X and Y: player position on the field

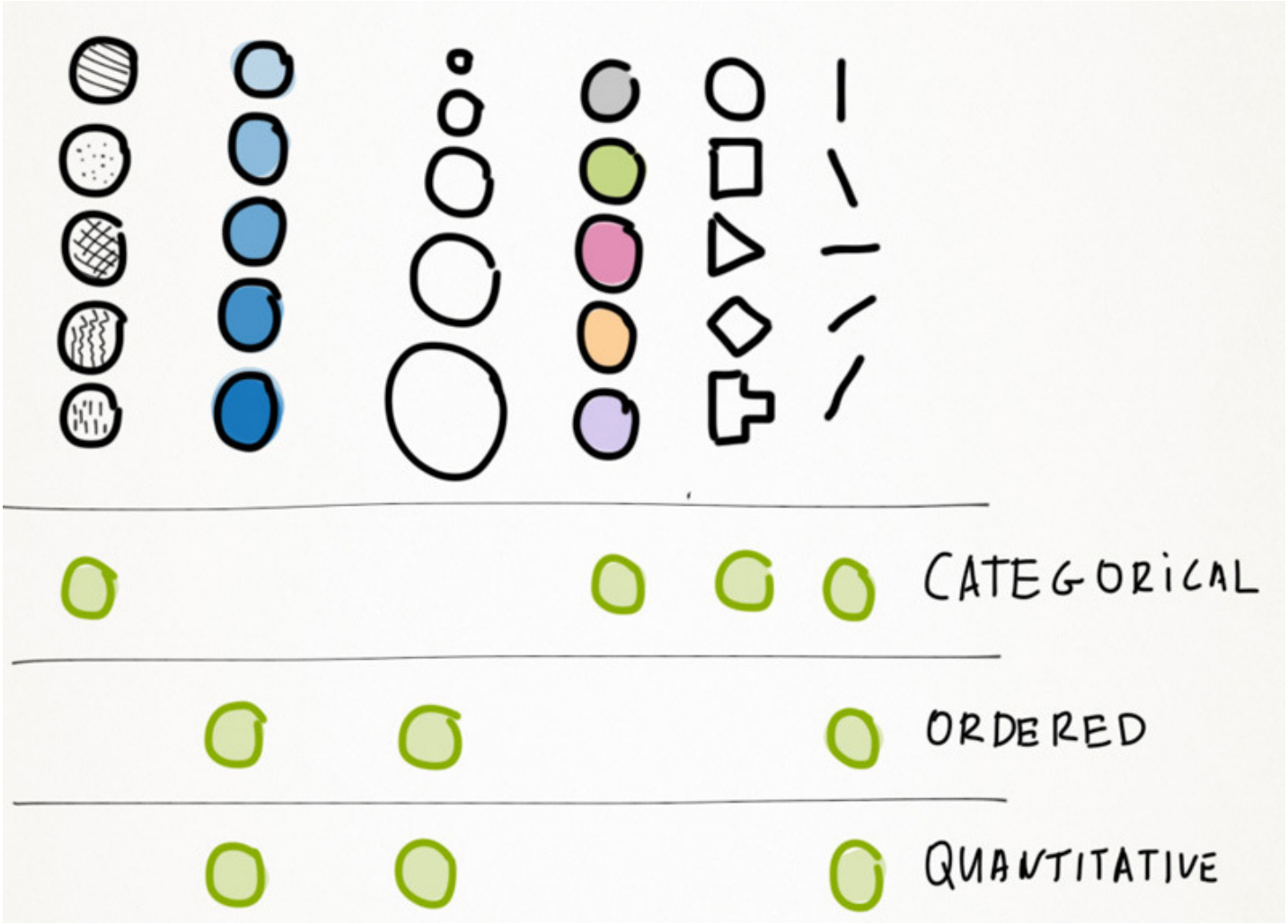
#8. Shoes

A very complex and somewhat crazy example where all the retinal variables are used: the shoes wall. Take a look at those diverse visualizations. The concept can be quite hard to grasp, but it's curiosity that should be driving us to explore the details.



Wrap-up

The encoding variables power clear and intelligent visualizations when used wisely. Avoid common mistakes, identify your data types and pick the relevant variables. It's easy.



Crafted by Michael Dubakov and Olga Kouzina

Spread the word:

Other articles by Michael Dubakov

The Future of Agile Software Development
Patterns for Information Visualization
Our Development Process: 50 Months of Evolution