

# Welcome to the tutorial!

**Marks & Channels**  
**Tabular Data**

Syed Affan Daimi



# QR Code

**Code  
Notebook**



**Optional**

# Agenda



1.

Recap: Marks and Channels



2.

Scatterplots

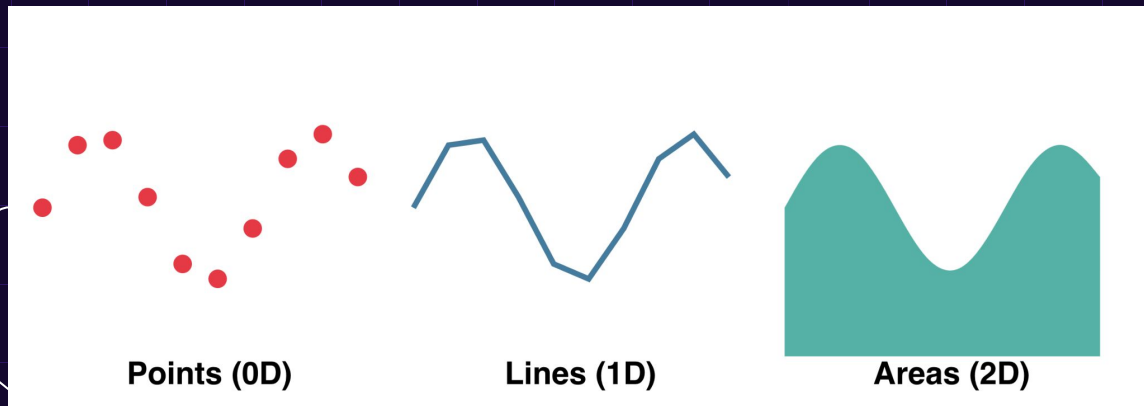


3.

Can we do better?

# Marks & Channels

Nouns



`<circle>`  
`<rect>` (small square)

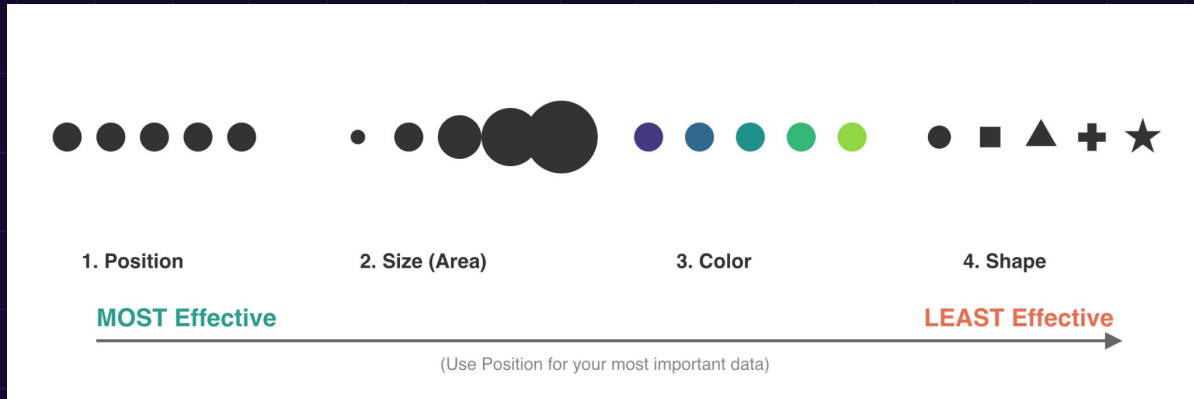
`<line>`, `<path>`

`<rect>`, `<path>`, `<polygon>`



# Marks & Channels

Adjectives



# Marks and Channels

```
const data = [  
  { country: "Norway",      cont: "Europe",  y1955: 72, y2015: 82 },  
  { country: "United States", cont: "Americas", y1955: 68, y2015: 79 },  
  { country: "South Korea",  cont: "Asia",    y1955: 47, y2015: 82 },  
  { country: "Vietnam",      cont: "Asia",    y1955: 45, y2015: 76 },  
  { country: "Brazil",       cont: "Americas", y1955: 55, y2015: 75 },  
  { country: "China",        cont: "Asia",    y1955: 44, y2015: 76 },  
  { country: "Egypt",        cont: "Africa",   y1955: 42, y2015: 71 },  
  { country: "India",        cont: "Asia",    y1955: 38, y2015: 68 },  
  { country: "Nigeria",     cont: "Africa",   y1955: 37, y2015: 54 },  
  { country: "Sierra Leone", cont: "Africa",   y1955: 30, y2015: 51 }  
];
```

**Rows**

**Columns**

# Marks

```
{  
  // ... svg setup ...  
  svg.selectAll("circle")  
    .data(data)  
    .join("circle")  
    .attr("r", 8)  
    .attr("fill", "steelblue");  
  // ...  
}
```

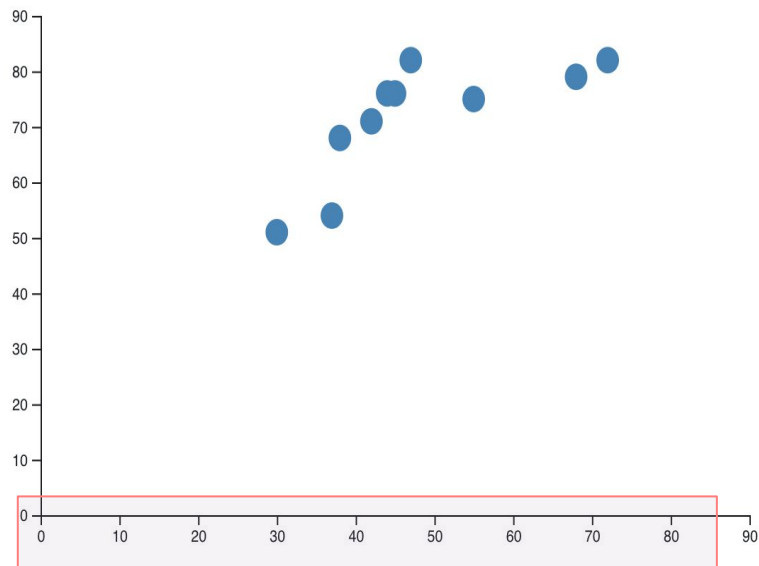
Create the **MARKS**  
(Geometric  
Primitives)

Bind our table rows

**Fixed Channel (For now)**



# Creating a Scatter Plot

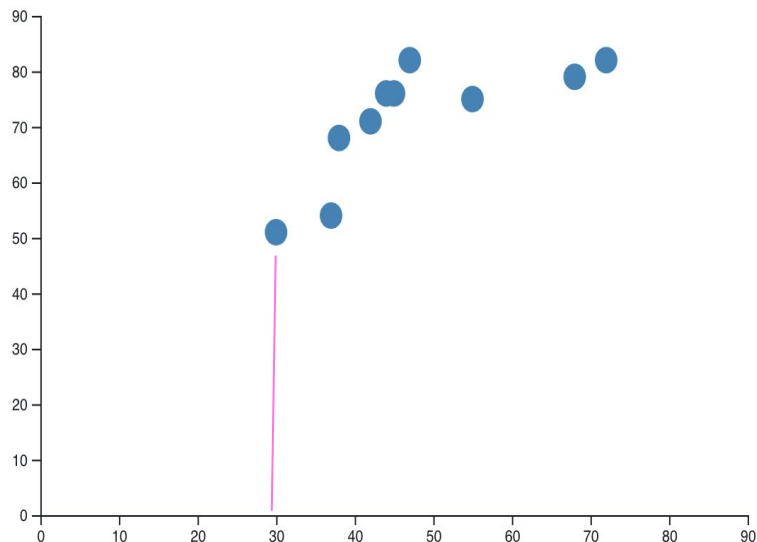


```
{  
  // ... svg setup ...  
  const x = d3.scaleLinear()  
    .domain([0, 90])  
    .range([50, 550]);  
  
  const y = d3.scaleLinear()  
    .domain([0, 90])  
    .range([350, 50]);  
  // ...  
}
```





# Creating a Scatter Plot



```
{  
  // ... scales defined above ...  
  svg.selectAll("circle")  
    .data(data)  
    .join("circle")  
    .attr("cx", d => x(d.y1955))  
    .attr("cy", d => y(d.y2015));  
  // ...  
}
```

**// CHANNEL:**

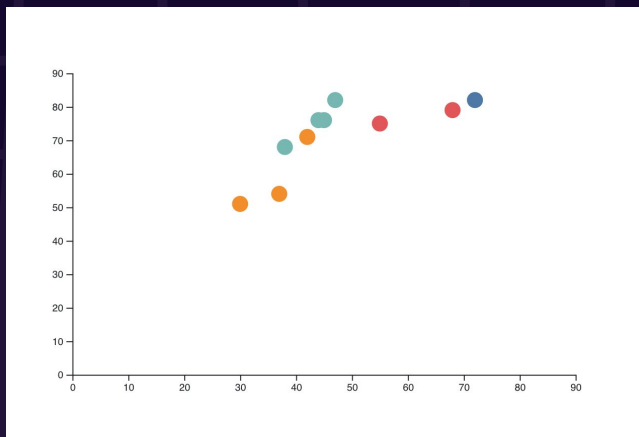
**Express Value 1  
(1955)**

**CHANNEL: Express Value 2  
(2015)**





# Scatter Plot: Adding Colors

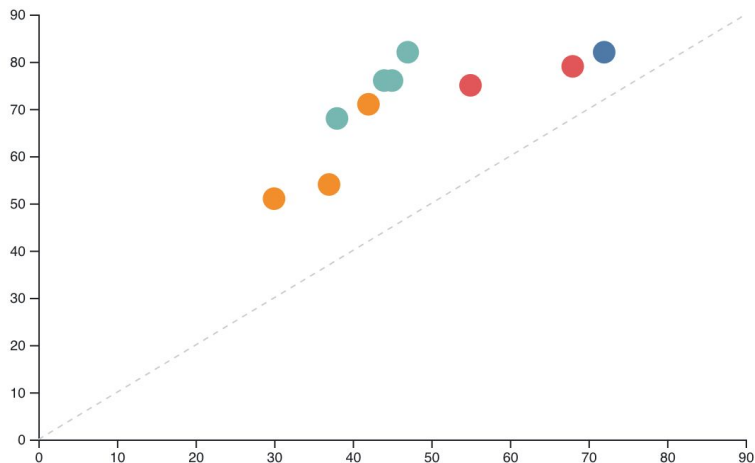


```
{  
  // ...  
  const color = d3.scaleOrdinal()  
    .domain(["Europe", "Americas", "Asia",  
            "Africa"])  
    .range(["#4e79a7", "#e15759", "#76b7b2",  
            "#f28e2b"]);  
  
  svg.selectAll("circle")  
    // ... position attributes ...  
    .attr("fill", d => color(d.cont));  
}
```

Mapping  
categorical  
information  
into colors



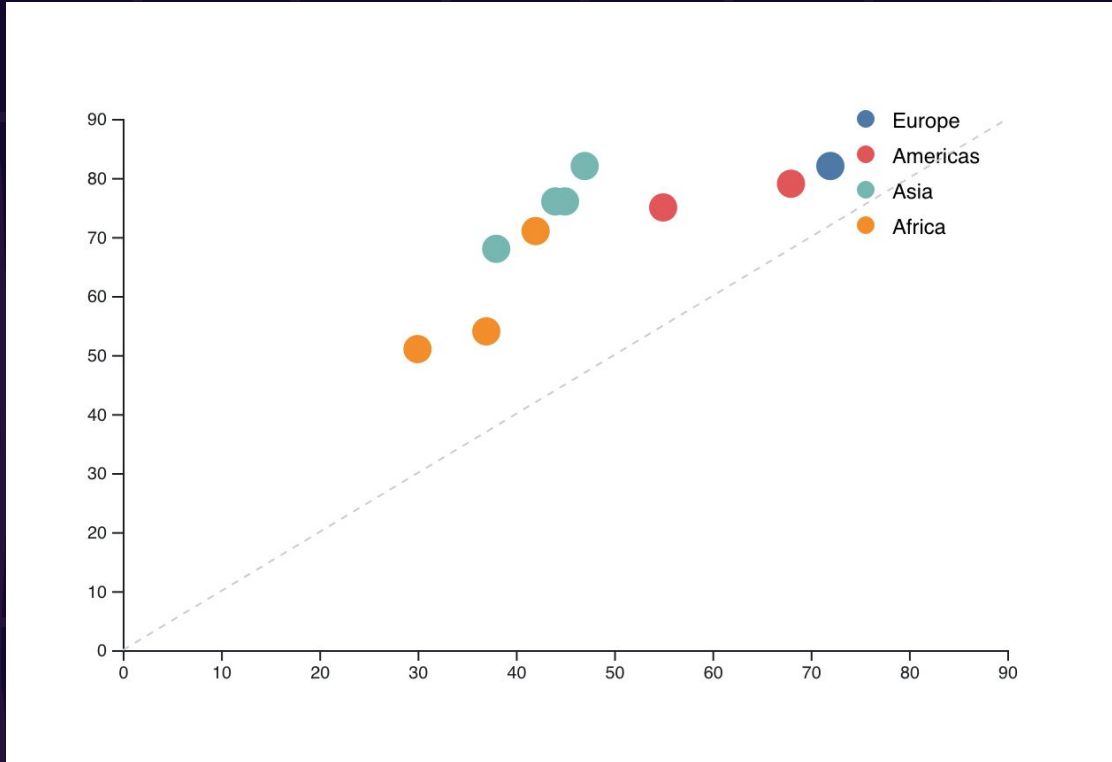
# Scatter Plot: Adding Colors



```
{  
  // ... scatter plot code ...  
  
  // Add Reference Mark  
  svg.append("line")  
    .attr("x1", x(0)).attr("y1", y(0)) // Start (0,0)  
    .attr("x2", x(90)).attr("y2", y(90)) // End (90,90)  
    .attr("stroke", "#ccc")  
    .attr("stroke-dasharray", "4"); // Dashed  
}
```



# Final Product



**Now what?**

How can we compare life expectancies from this?



# Back to the Marks



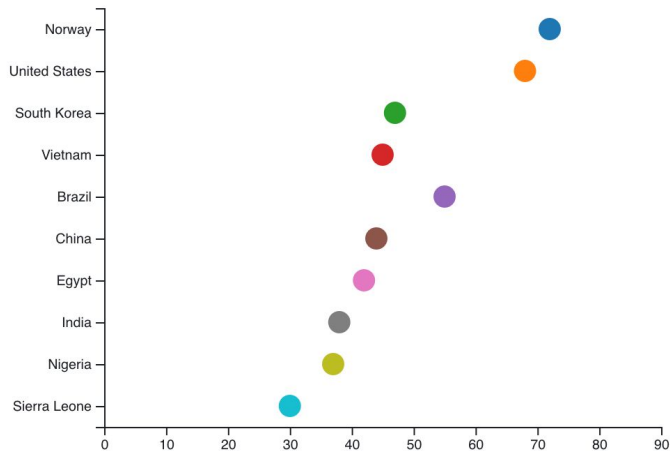
```
{  
  const yTable = d3.scalePoint()  
    .domain(data.map(d => d.country))  
    .range([50, 350]);  
  
  svg.selectAll("circle")  
    .data(data).join("circle")  
    .attr("cx", 100) // No value yet  
    .attr("cy", d => yTable(d.country));  
}
```

**Y-Scale is  
now  
categorical**





# Back to the Marks

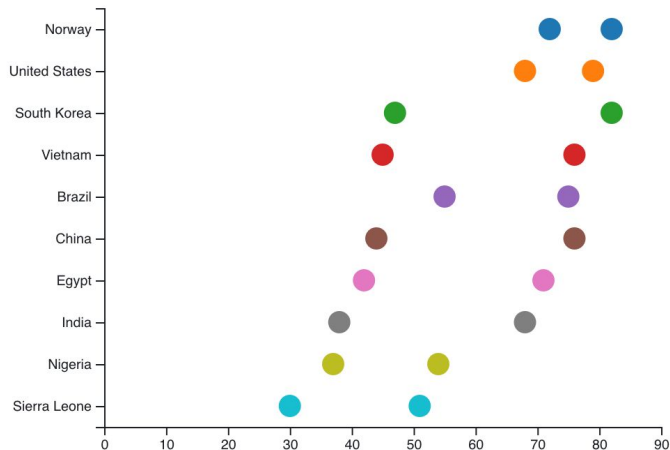


```
{  
  // ... axis code ...  
  svg.selectAll("circle")  
    .data(data).join("circle")  
  
  // ACTION: Express Value (1955)  
  .attr("cx", d => x(d.y1955))  
  
  .attr("cy", d => yTable(d.country));  
}
```





# Back to the Marks



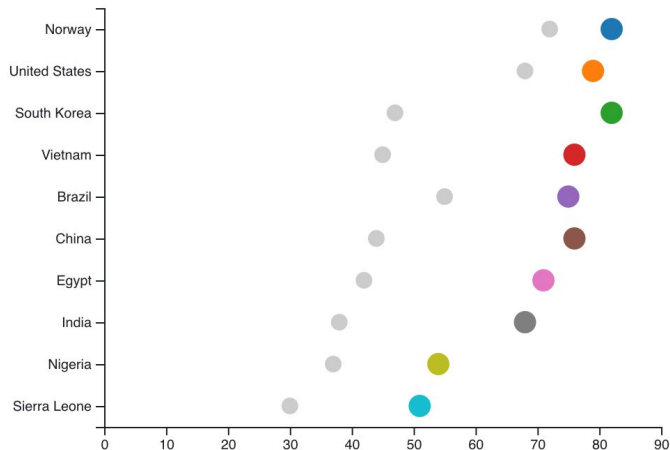
```
{  
  // 1955 Dots  
  svg.selectAll("circle.start").data(data).join("circle")  
    .attr("cx", d => x(d.y1955))...  
  
  // 2015 Dots  
  svg.selectAll("circle.end").data(data).join("circle")  
    .attr("cx", d => x(d.y2015))...  
}
```

**PROBLEM:** Ambiguity.  
Which is which?





# Back to the Marks



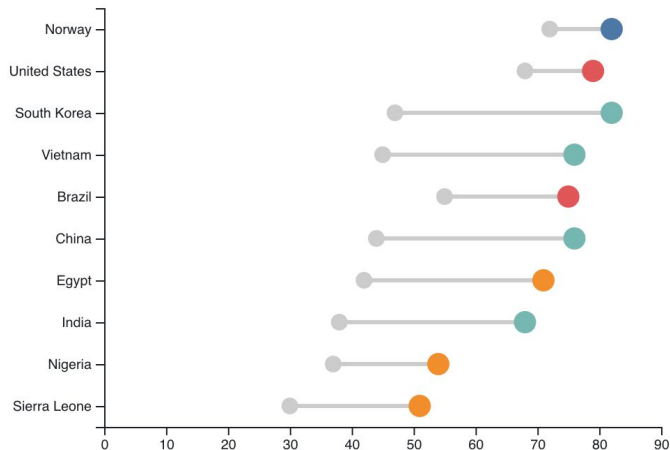
```
{  
  // 1955 Dots (Context)  
  svg.selectAll("circle.start")  
    .attr("fill", "#ccc"); // Low Salience  
  
  // 2015 Dots (Focus)  
  svg.selectAll("circle.end")  
    .attr("fill", d => color(d.cont));  
}
```

Encode Continent  
(Identity)





# Back to the Marks

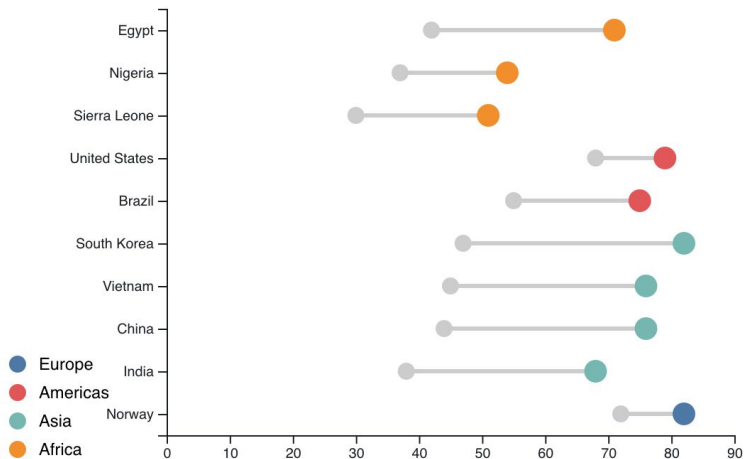


```
svg.selectAll("line.connect")  
  .data(data)  
  .join("line")  
  .attr("class", "connect")  
  .attr("x1", d => x(d.y1955))  
  .attr("x2", d => x(d.y2015))  
  .attr("y1", d => yTable(d.country))  
  .attr("y2", d => yTable(d.country))  
  .attr("stroke", "#ccc").attr("stroke-width", 3);
```





# Final Product (Again)



```
{  
  // Option A: Order by Continent (Grouping)  
  data.sort((a, b) => d3.ascending(a.cont, b.cont));  
  
  // Option B: Order by Value (Ranking)  
  data.sort((a, b) => b.y2015 - a.y2015);  
  
  // Re-run the Scale Domain  
  yTable.domain(data.map(d => d.country));  
}
```





# Questions?

Are there any questions?