unesco
University Network
Member

Ministry of
Education

Effective Visualization

Presented by Hyebong Choi

TOT

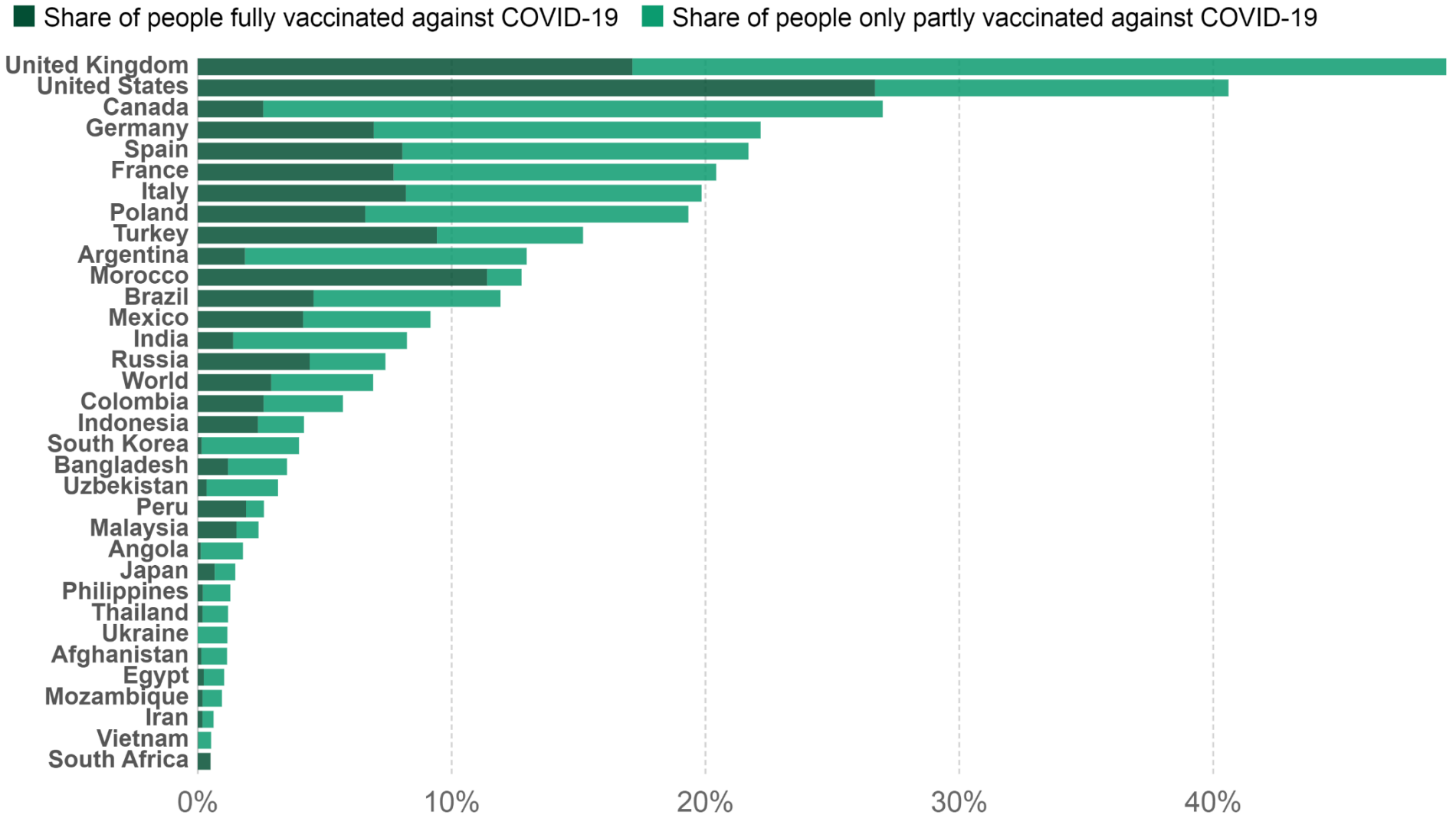
Data Visualization

- Essential component of skill set as a data scientist
- With ever increasing volume of data, it is impossible to tell stories without visualizations.
- Data visualization is an art of how to turn “Big Data” into useful knowledge

Share of people vaccinated against COVID-19, Apr 22, 2021

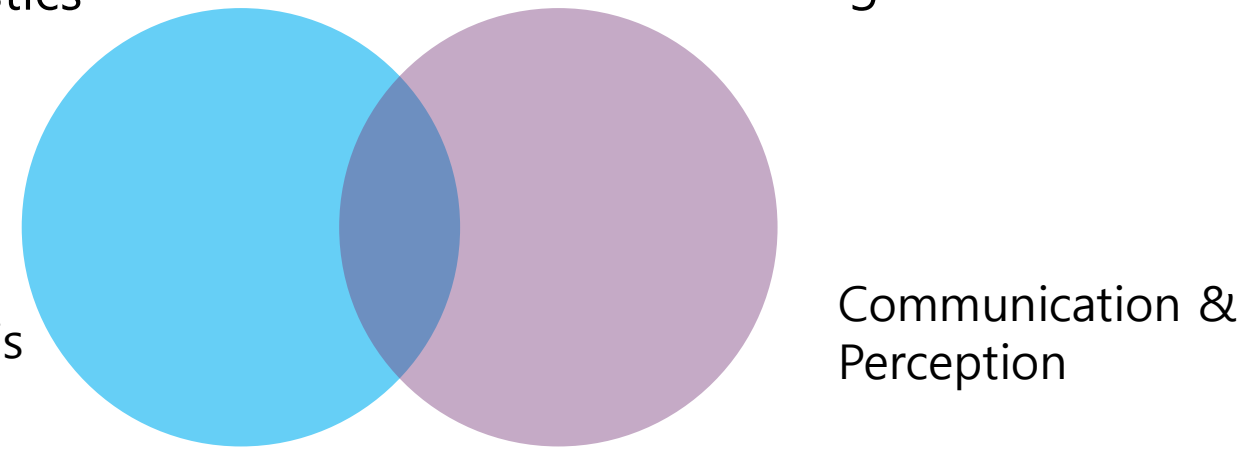
This data is only available for countries which report the breakdown of doses administered by first and second doses.

Our World in Data



Source: Official data collated by Our World in Data

Data Visualization



2 Different Main Purposes

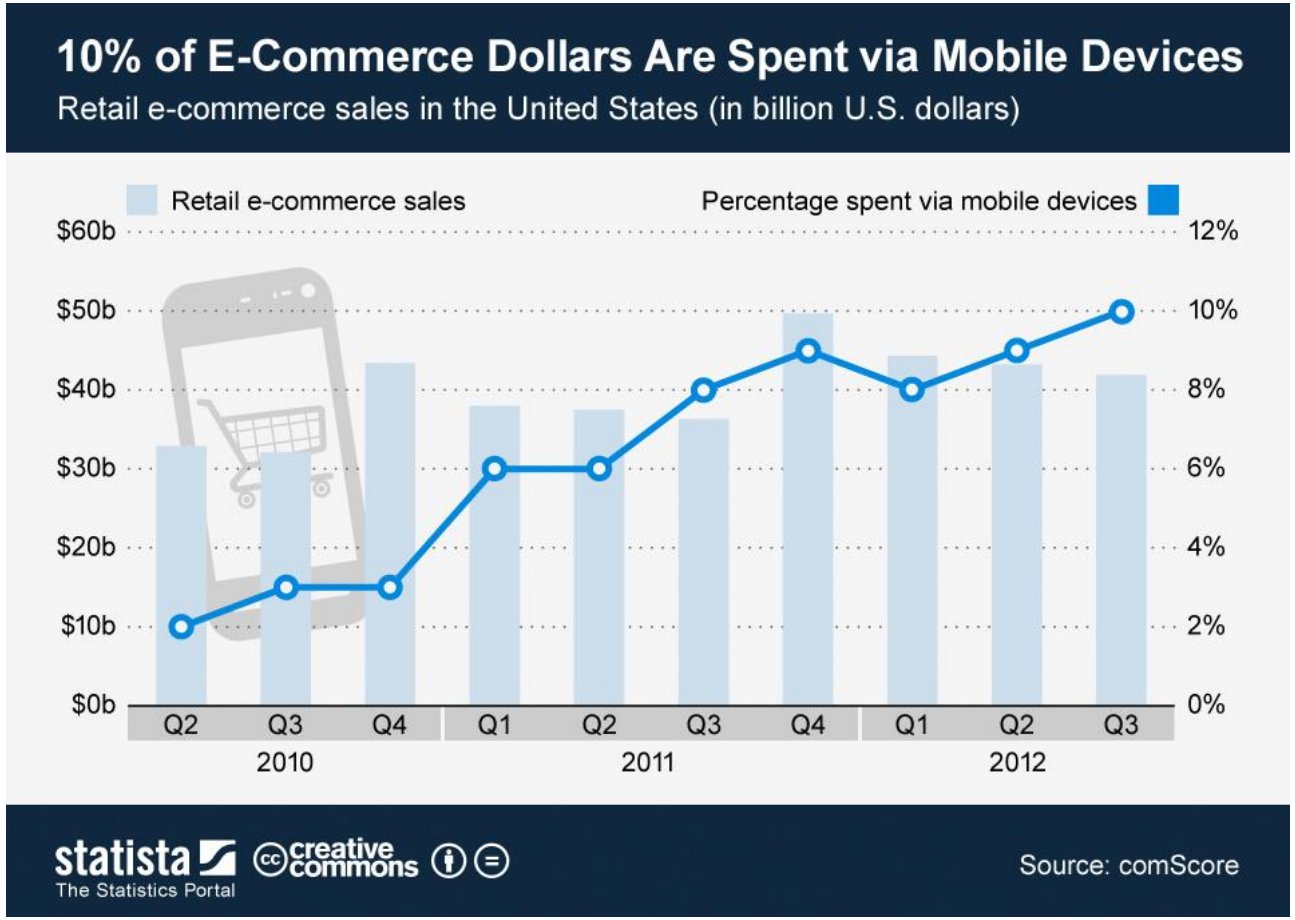
- Exploratory Visualization
  - Help you see what is in the data
- Explanatory Visualization
  - Shows others what you've found in your data

2 Different Main Purposes

- Exploratory Visualization
  - Help you see what is in the data
  - Keep as much as detail as possible
  - Practical Limit: how much can you see and interpret
- Explanatory Visualization
  - Help us share our understanding with others
  - Shows others what you've found in your data
  - Requires editorial decisions:
    - Highlight the key features you want to emphasize
    - Eliminate extraneous details

Exploratory Visualization

Explanatory Visualization



Google Chart

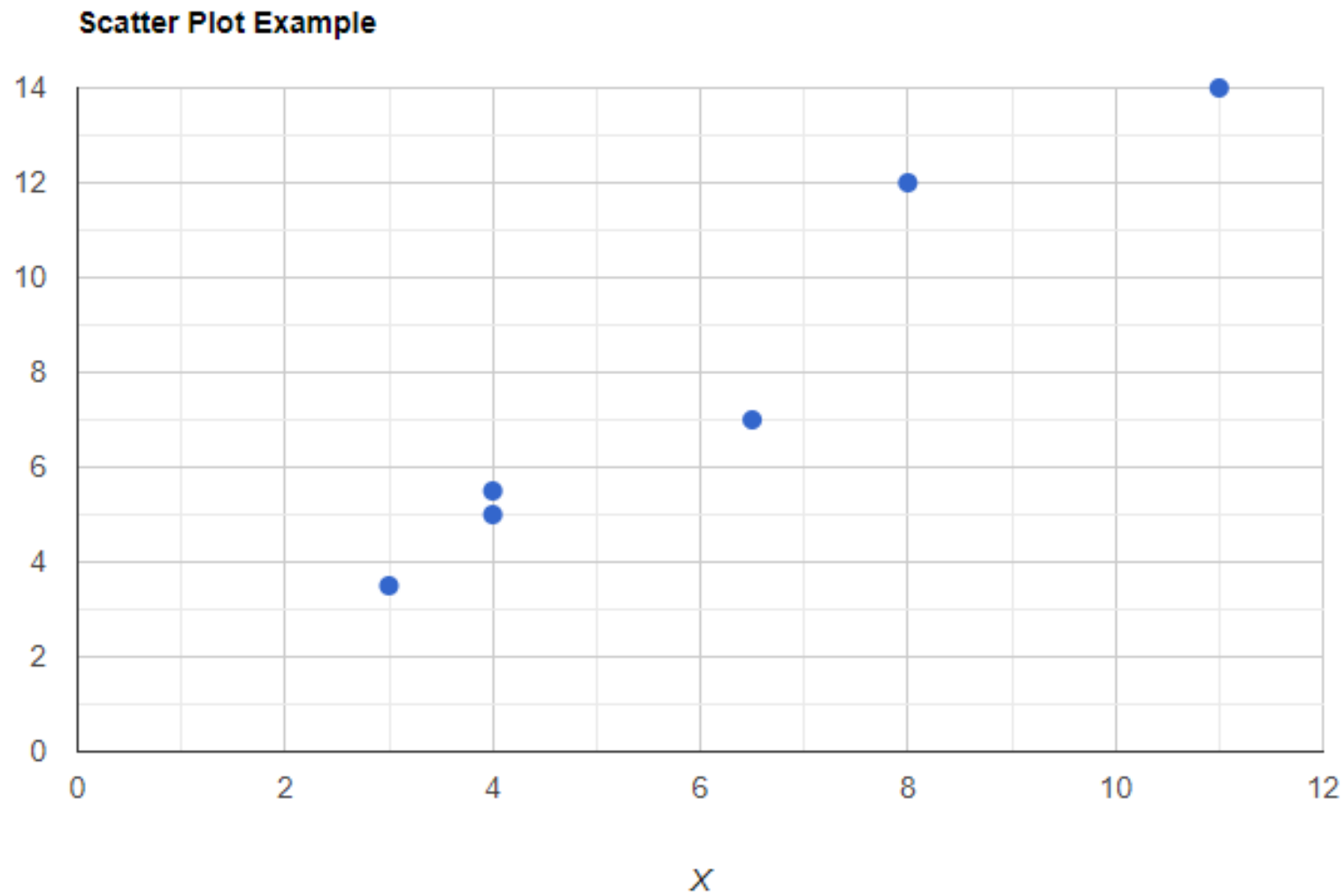
First trial

Basic Setup

```
<!DOCTYPE html>
<html>
    <head>
        <script type="text/javascript" src="https://www.gstatic.com/charts/loader.js"></script>
        <script type="text/javascript">
            google.charts.load('current', {'packages':['corechart']});
            google.charts.setOnLoadCallback(drawChart);
            function drawChart() {
            // Chart code will go here
        }
        </script>
    </head>
    <body>
        <div id="chart_div"></div>
    </body>
</html>
```

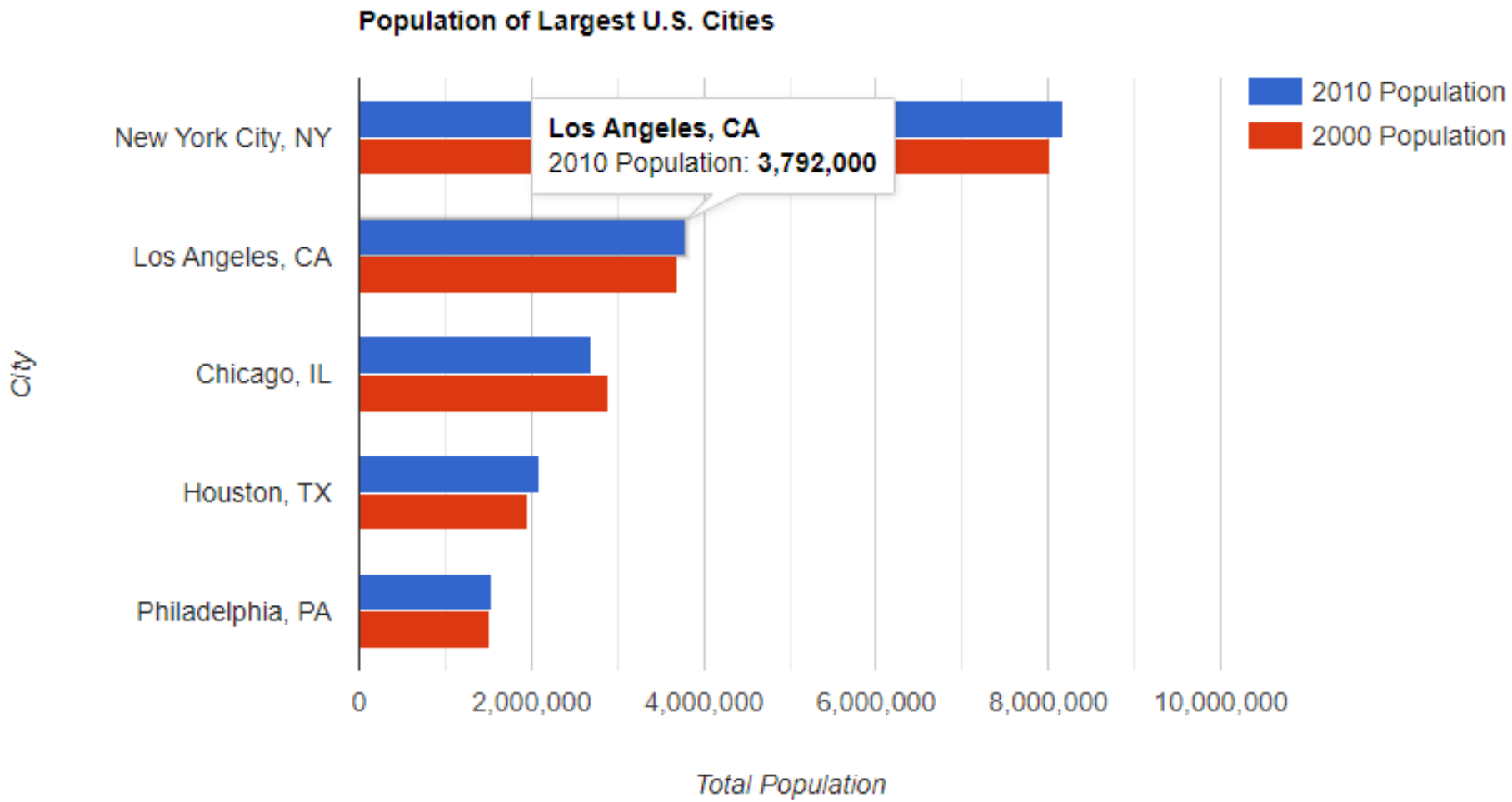
Creating a Scatter Plot

```
<!DOCTYPE html>
<html>
	<head>
		<script type="text/javascript" src="https://www.gstatic.com/charts/loader.js"></script>
		<script type="text/javascript">
			google.charts.load('current', {'packages':['corechart']});
			google.charts.setOnLoadCallback(drawChart);
			function drawChart() {
				var data = google.visualization.arrayToDataTable([
				['X', 'Y'],
				[8, 12],
				[4, 5.5],
				[11, 14],
				[4, 5],
				[3, 3.5],
				[6.5, 7]
			]);
			var options = {
				title: 'Scatter Plot Example',
				hAxis: {title: 'X'},
				vAxis: {title: 'Y'},
				legend: 'none'
			};
			var chart = new google.visualization.ScatterChart(document.getElementById('chart_div'));
			chart.draw(data, options);
			}
		</script>
	</head>
	<body>
		<div id="chart_div" style="width: 900px; height: 500px;"></div>
	</body>
</html>
```


Scatter Plot Example
14
12
10
8
6
4
2
0
0
2
4
6
8
10
12
X

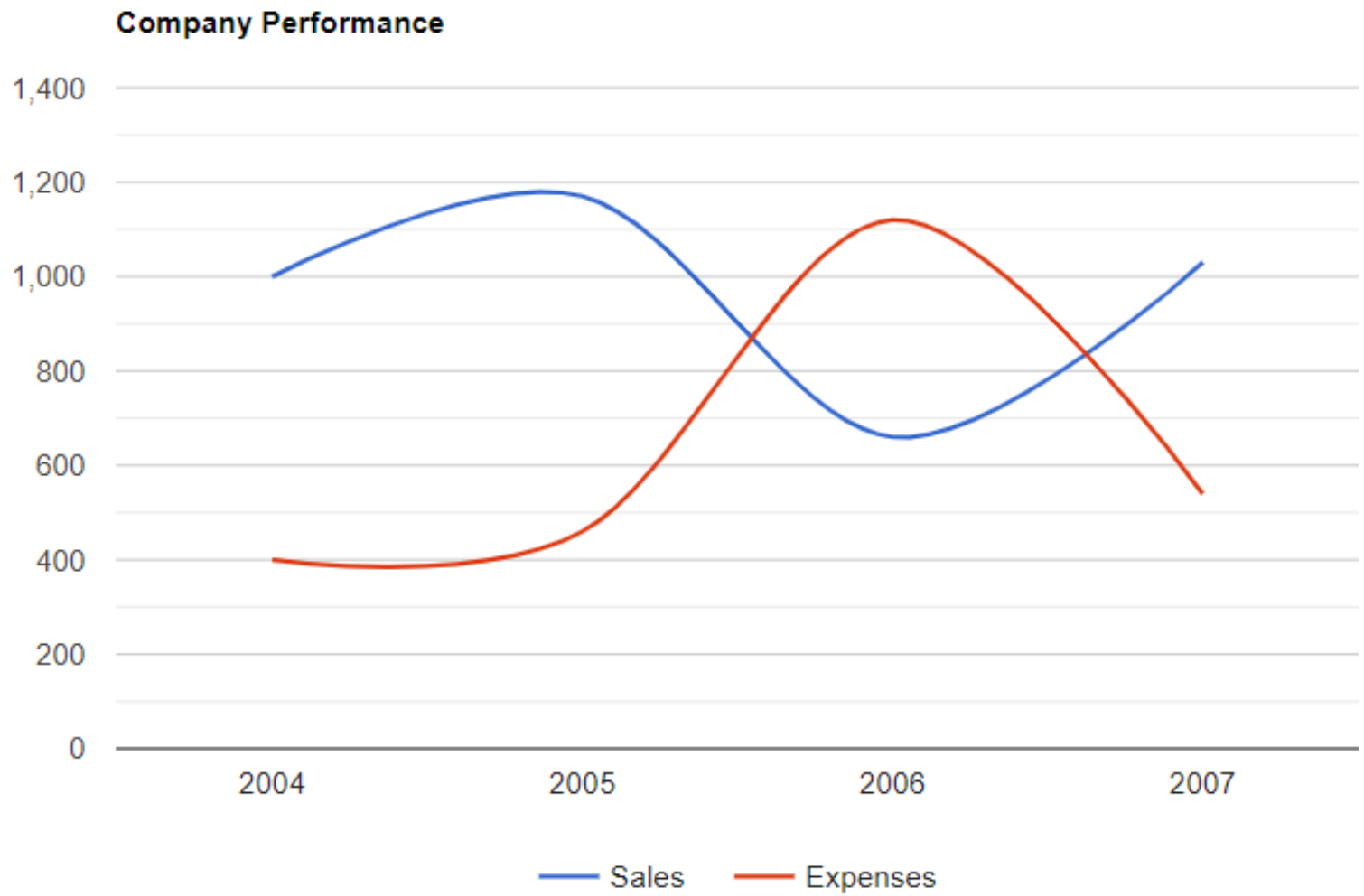
Creating a Bar Chart

```html
<!DOCTYPE html>
<html>
  <head>
    <script type="text/javascript" src="https://www.gstatic.com/charts/loader.js"></script>
    <script type="text/javascript">
      google.charts.load('current', {'packages':['corechart']});
      google.charts.setOnLoadCallback(drawChart);

      function drawChart() {
        var data = google.visualization.arrayToDataTable([
          ['City', '2010 Population', '2000 Population'],
          ['New York City, NY', 8175000, 8008000],
          ['Los Angeles, CA', 3792000, 3694000],
          ['Chicago, IL', 2695000, 2896000],
          ['Houston, TX', 2099000, 1953000],
          ['Philadelphia, PA', 1526000, 1517000]
        ]);

        var options = {
          title: 'Population of Largest U.S. Cities',
          chartArea: {width: '50%'},
          hAxis: {
            title: 'Total Population',
            minValue: 0
          },
          vAxis: {
            title: 'City'
          }
        };

        var chart = new google.visualization.BarChart(document.getElementById('chart_div'));
        chart.draw(data, options);
      }
    </script>
  </head>
  <body>
    <div id="chart_div" style="width: 900px; height: 500px;"></div>
  </body>
</html>
```


Population of Largest U.S. Cities
2010 Population
2000 Population
Los Angeles, CA
2010 Population: 3,792,000
New York City, NY
Los Angeles, CA
Chicago, IL
Houston, TX
Philadelphia, PA
City
0
2,000,000
4,000,000
6,000,000
8,000,000
10,000,000
Total Population

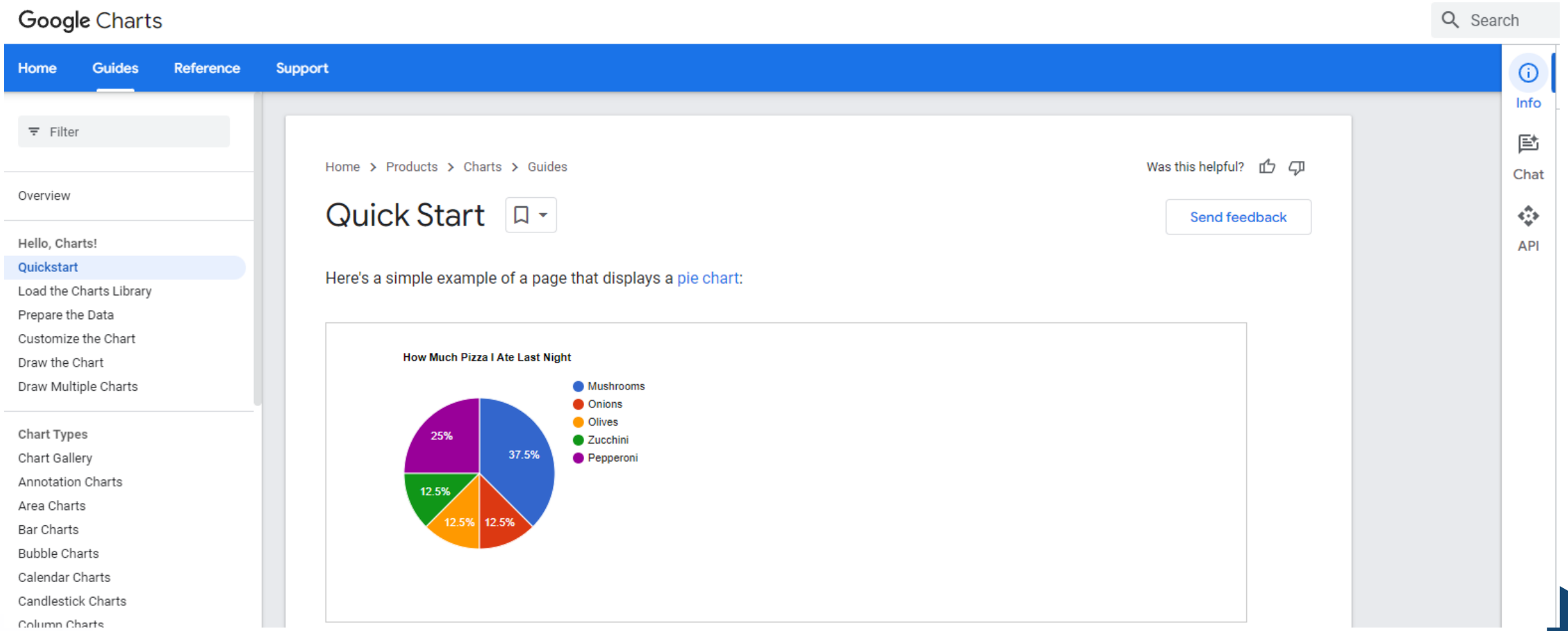
Creating a Line plot

```html
<!DOCTYPE html>
<html>
  <head>
    <script type="text/javascript" src="https://www.gstatic.com/charts/loader.js"></script>
    <script type="text/javascript">
      google.charts.load('current', {'packages':['corechart']});
      google.charts.setOnLoadCallback(drawChart);

      function drawChart() {
        var data = google.visualization.arrayToDataTable([
          ['Year', 'Sales', 'Expenses'],
          ['2004',  1000,      400],
          ['2005',  1170,      460],
          ['2006',  660,       1120],
          ['2007',  1030,      540]
        ]);

        var options = {
          title: 'Company Performance',
          curveType: 'function',
          legend: { position: 'bottom' }
        };

        var chart = new google.visualization.LineChart(document.getElementById('chart_div'));
        chart.draw(data, options);
      }
    </script>
  </head>
  <body>
    <div id="chart_div" style="width: 900px; height: 500px;"></div>
  </body>
</html>
```


Company Performance
1,400
1,200
1,000
800
600
400
200
0
2004
2005
2006
2007
Sales
Expenses

Try some tutorials here

https://developers.google.com/chart/interactive/docs/quick_start



Loading Dataset from files

First, let's assume you have a CSV file named `data.csv` with the following content:

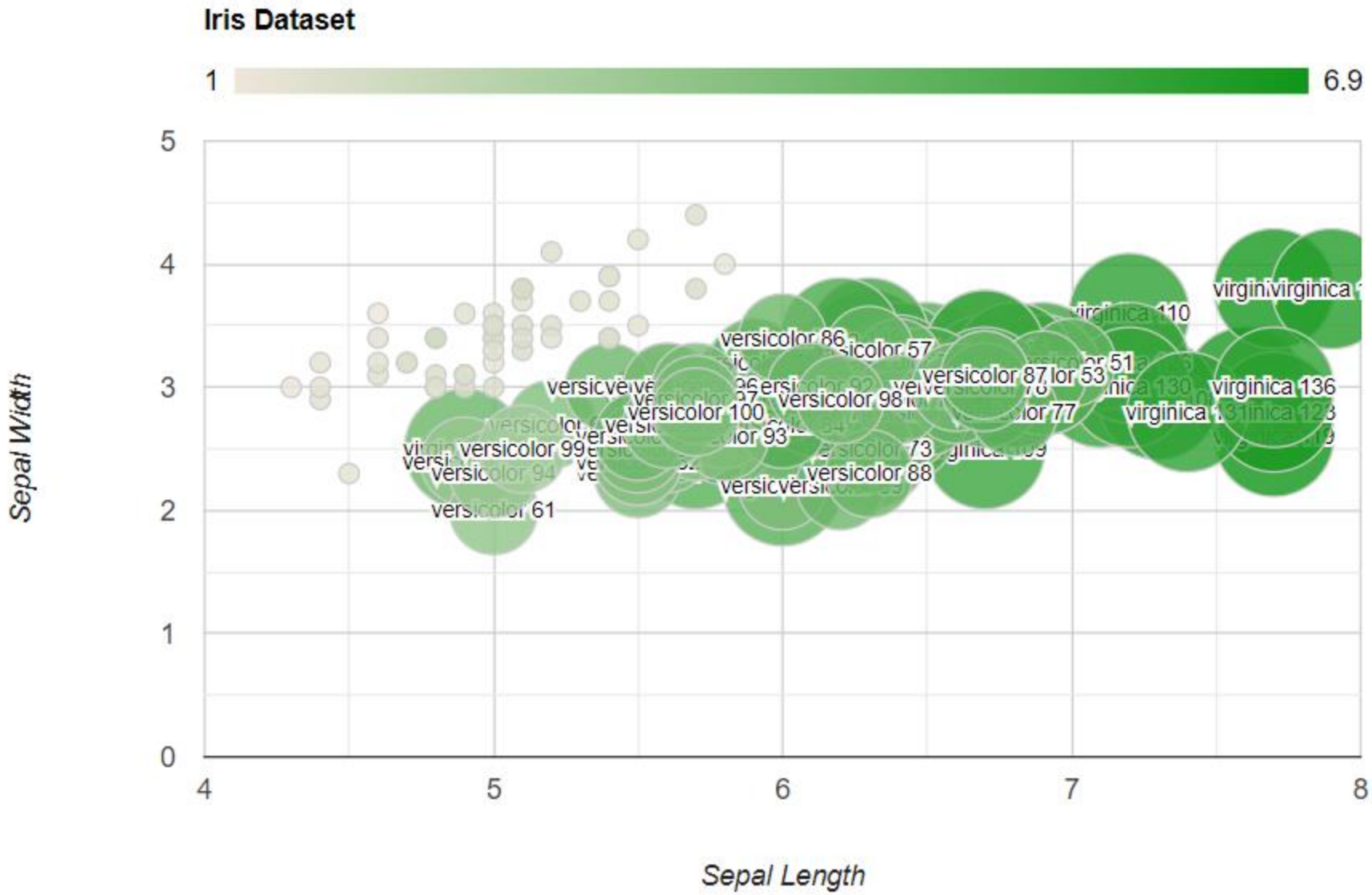
yaml Copy code

```yaml
Year,Sales,Expenses
2004,1000,400
2005,1170,460
2006,660,1120
2007,1030,540
```

```html
<html>
<head>
    <title>Google Chart with CSV Data</title>
    <script type="text/javascript" src="https://www.gstatic.com/charts/loader.js"></script>
    <script src="https://d3js.org/d3.v5.min.js"></script>
</head>
<body>
    <div id="chart_div" style="width: 900px; height: 500px;"></div>

    <script type="text/javascript">
        // Load Google Charts library
        google.charts.load('current', {'packages':['corechart']});
        google.charts.setOnLoadCallback(drawChart);

        // Function to draw the chart
        function drawChart() {
            // Load CSV data from a web URL
            d3.csv('https://raw.githubusercontent.com/mwaskom/seaborn-data/master/iris.csv').then(function(data) {
                // Map species to numeric values
                const speciesMap = {
                    'setosa': 1,
                    'versicolor': 2,
                    'virginica': 3
                };

                // Parse the CSV data
                const dataArray = [['ID', 'Sepal Length', 'Sepal Width', 'Petal Length', 'Species (Numeric)']];
                data.forEach(function(row, index) {
                    dataArray.push([
                        row['species'] + ' ' + (index + 1),
                        parseFloat(row['sepal_length']),
                        parseFloat(row['sepal_width']),
                        parseFloat(row['petal_length']),
                        speciesMap[row['species']]
                    ]);
                });

                // Convert data to Google Charts DataTable format
                const googleData = google.visualization.arrayToDataTable(dataArray);

                // Set chart options
                const options = {
                    title: 'Iris Dataset',
                    hAxis: {title: 'Sepal Length'},
                    vAxis: {title: 'Sepal Width'},
                    bubble: {textStyle: {fontSize: 11}}
                };

                // Draw the chart
                const chart = new google.visualization.BubbleChart(document.getElementById('chart_div'));
                chart.draw(googleData, options);
            }).catch(function(error) {
                console.error('Error loading or parsing CSV:', error);
            });
        }
    </script>
</body>
</html>
```

Species	SepalLeng	SepalWidt	PetalLeng	PetalWidt
setosa	5.1 cm	3.5 cm	1.4 cm	0.2 cm
setosa	4.9 cm	3. cm	1.4 cm	0.2 cm
setosa	4.7 cm	3.2 cm	1.3 cm	0.2 cm
setosa	4.6 cm	3.1 cm	1.5 cm	0.2 cm
setosa	5. cm	3.6 cm	1.4 cm	0.2 cm
setosa	5.4 cm	3.9 cm	1.7 cm	0.4 cm
setosa	4.6 cm	3.4 cm	1.4 cm	0.3 cm
setosa	5. cm	3.4 cm	1.5 cm	0.2 cm
setosa	4.4 cm	2.9 cm	1.4 cm	0.2 cm
setosa	4.9 cm	3.1 cm	1.5 cm	0.1 cm
setosa	5.4 cm	3.7 cm	1.5 cm	0.2 cm
setosa	4.8 cm	3.4 cm	1.6 cm	0.2 cm
setosa	4.8 cm	3. cm	1.4 cm	0.1 cm
setosa	4.3 cm	3. cm	1.1 cm	0.1 cm
setosa	5.8 cm	4. cm	1.2 cm	0.2 cm
setosa	5.7 cm	4.4 cm	1.5 cm	0.4 cm
setosa	5.4 cm	3.9 cm	1.3 cm	0.4 cm
setosa	5.1 cm	3.5 cm	1.4 cm	0.3 cm
setosa	5.7 cm	3.8 cm	1.7 cm	0.3 cm
setosa	5.1 cm	3.8 cm	1.5 cm	0.3 cm

rows 1–20 of **150**

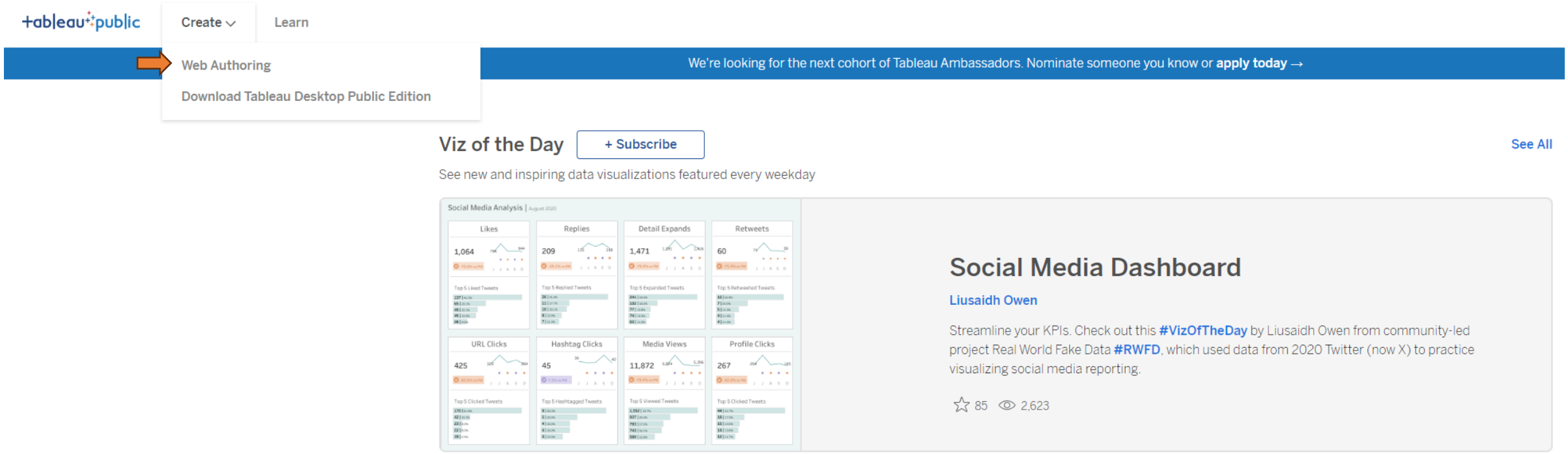

Iris Dataset
1
6.9
5
4
3
2
1
0
4
5
6
7
8
Sepal Width
Sepal Length
versicolor 61
versicolor 99
versicolor 100
versicolor 93
versicolor 98
versicolor 86
versicolor 57
versicolor 87
versicolor 88
virginica 110
virginica 136
virginica 123

Tableau

First trial

Tableau Public

https://public.tableau.com/app/discover



Sample Dataset for trial

Restaurant Revenue Prediction Dataset

A dataset about restaurant

Data Card | Code (0) | Discussion (0) | Suggestions (0)

About Dataset

This dataset contains information about various restaurants and aims to predict the revenue based on several features. Each row represents a unique restaurant with various attributes that may influence its revenue.

Columns

- **Name**: The name of the restaurant.
- **Location**: The location of the restaurant (e.g., Rural, Downtown).
- **Cuisine**: The type of cuisine offered (e.g., Japanese, Mexican, Italian).
- **Rating**: The average rating of the restaurant.
- **Seating Capacity**: The number of seats available in the restaurant.
- **Average Meal Price**: The average price of a meal at the restaurant.
- **Marketing Budget**: The marketing budget allocated for the restaurant.
- **Social Media Followers**: The number of social media followers.
- **Chef Experience Years**: The number of years of experience of the head chef.
- **Number of Reviews**: The total number of reviews the restaurant has received.
- **Avg Review Length**: The average length of reviews.
- **Ambience Score**: A score representing the ambience of the restaurant.
- **Service Quality Score**: A score representing the quality of service.
- **Parking Availability**: Indicates if parking is available (Yes/No).
- **Weekend Reservations**: The number of reservations made on weekends.
- **Weekday Reservations**: The number of reservations made on weekdays.
- **Revenue**: The total revenue generated by the restaurant.

Usability ⓘ
9.41

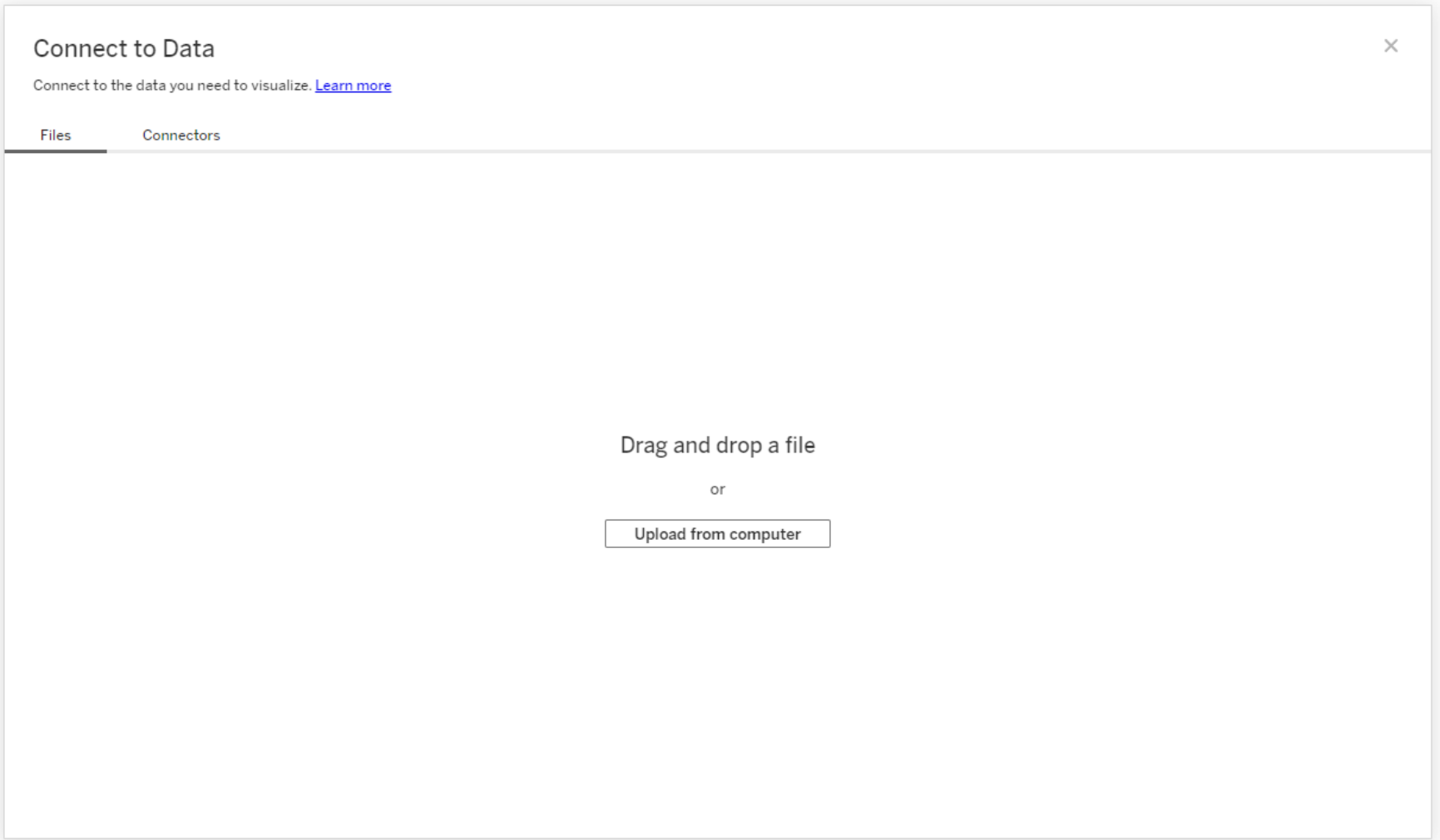
License
CC BY-SA 4.0

Expected update frequency
Never

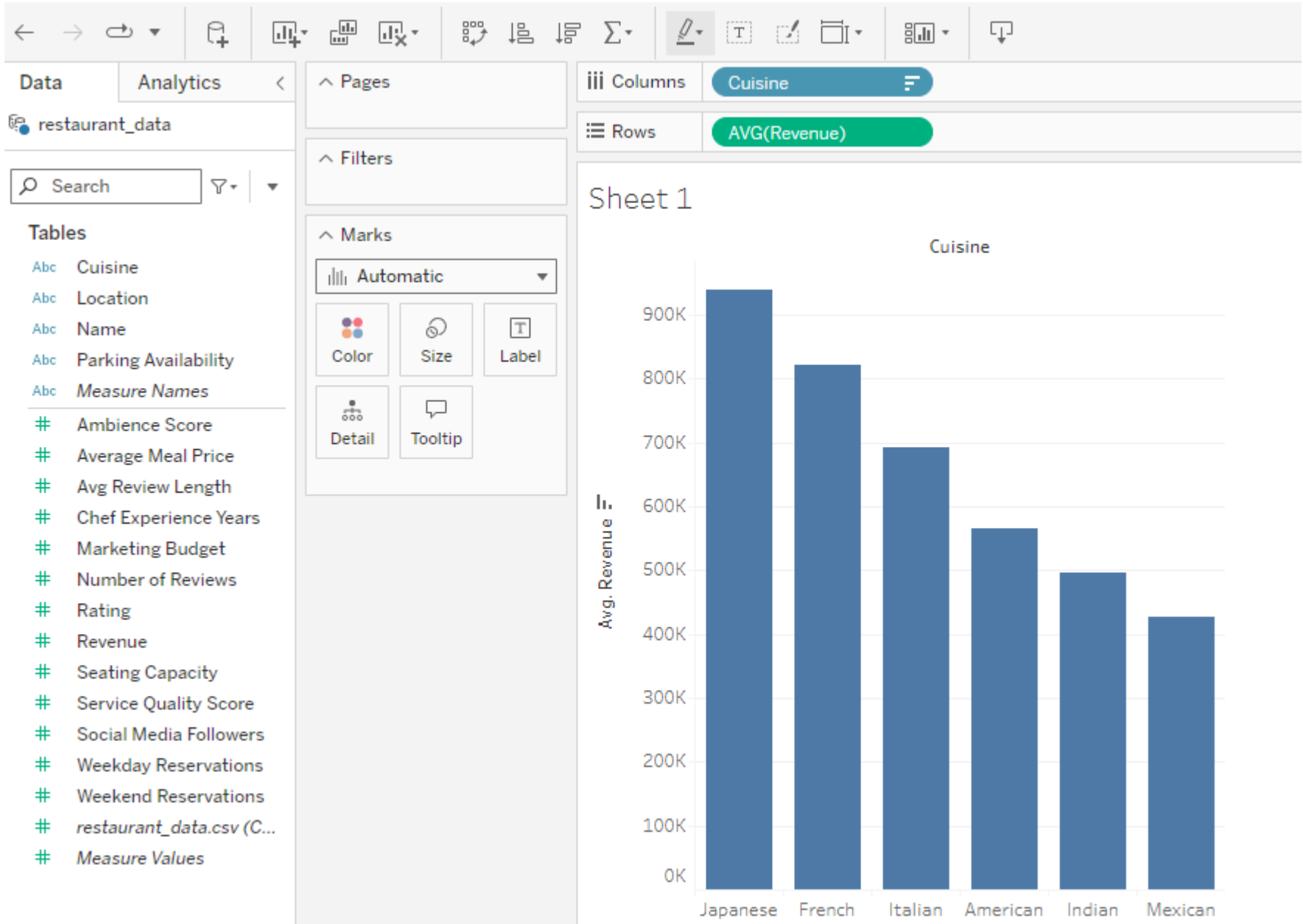
Tags

Tabular | Beginner | Data Analytics | Restaurants

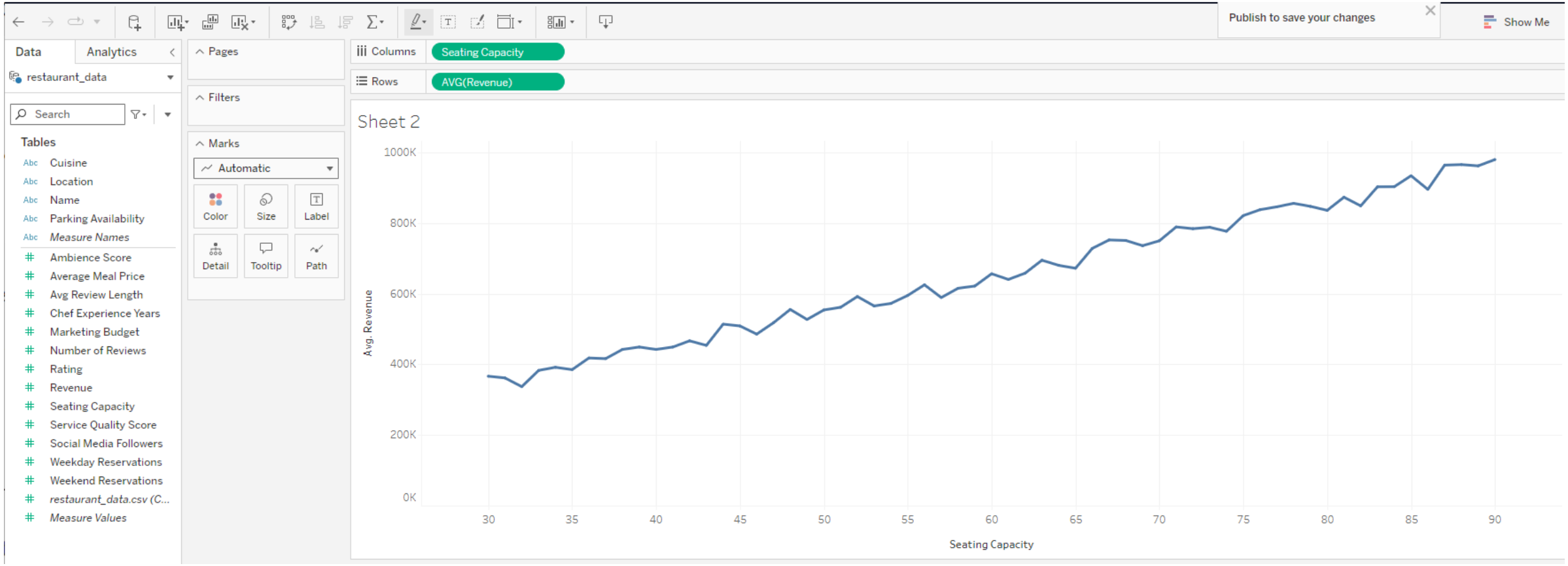
https://www.kaggle.com/datasets/anthonytherrien/restaurant-revenue-prediction-dataset


Connect to Data
Connect to the data you need to visualize. Learn more
Files
Connectors
Drag and drop a file
or
Upload from computer


restaurant_data
restaurant_data.csv


Data
Analytics
restaurant_data
Search
Tables
Abc Cuisine
Abc Location
Abc Name
Abc Parking Availability
Abc Measure Names
Ambience Score
Average Meal Price
Avg Review Length
Chef Experience Years
Marketing Budget
Number of Reviews
Rating
Revenue
Seating Capacity
Service Quality Score
Social Media Followers
Weekday Reservations
Weekend Reservations
restaurant_data.csv (C...
Measure Values
Pages
Filters
Marks
Automatic
Color
Size
Label
Detail
Tooltip
Columns
Cuisine
Rows
AVG(Revenue)
Sheet 1
Cuisine
Avg. Revenue
900K
800K
700K
600K
500K
400K
300K
200K
100K
0K
Japanese
French
Italian
American
Indian
Mexican


Publish to save your changes
Data
Analytics
restaurant_data
Search
Tables
Abc Cuisine
Abc Location
Abc Name
Abc Parking Availability
Abc Measure Names
Ambience Score
Average Meal Price
Avg Review Length
Chef Experience Years
Marketing Budget
Number of Reviews
Rating
Revenue
Seating Capacity
Service Quality Score
Social Media Followers
Weekday Reservations
Weekend Reservations
restaurant_data.csv (C...
Measure Values
Pages
Filters
Marks
Automatic
Color
Size
Label
Detail
Tooltip
Shape
Location
Columns
Weekend Reservations
Rows
Revenue
Sheet 1
1600K
1400K
1200K
1000K
800K
600K
400K
200K
Revenue
-5
0
5
10
15
20
25
30
35
40
45
50
55
60
65
70
75
80
85
90


Publish to save your changes
Show Me
Data
Analytics
restaurant_data
Search
Tables
Abc Cuisine
Abc Location
Abc Name
Abc Parking Availability
Abc Measure Names
Ambience Score
Average Meal Price
Avg Review Length
Chef Experience Years
Marketing Budget
Number of Reviews
Rating
Revenue
Seating Capacity
Service Quality Score
Social Media Followers
Weekday Reservations
Weekend Reservations
restaurant_data.csv (C...
Measure Values
Pages
Filters
Marks
Automatic
Color
Size
Label
Detail
Tooltip
Path
Columns
Seating Capacity
Rows
AVG(Revenue)
Sheet 2
1000K
800K
600K
400K
200K
0K
Avg. Revenue
30
35
40
45
50
55
60
65
70
75
80
85
90
Seating Capacity