

# Trendlines

## Overview

---

A trendline is a line superimposed on a chart revealing the overall direction of the data. Google Charts can automatically generate trendlines for Scatter Charts, Bar Charts, Column Charts, and Line Charts.

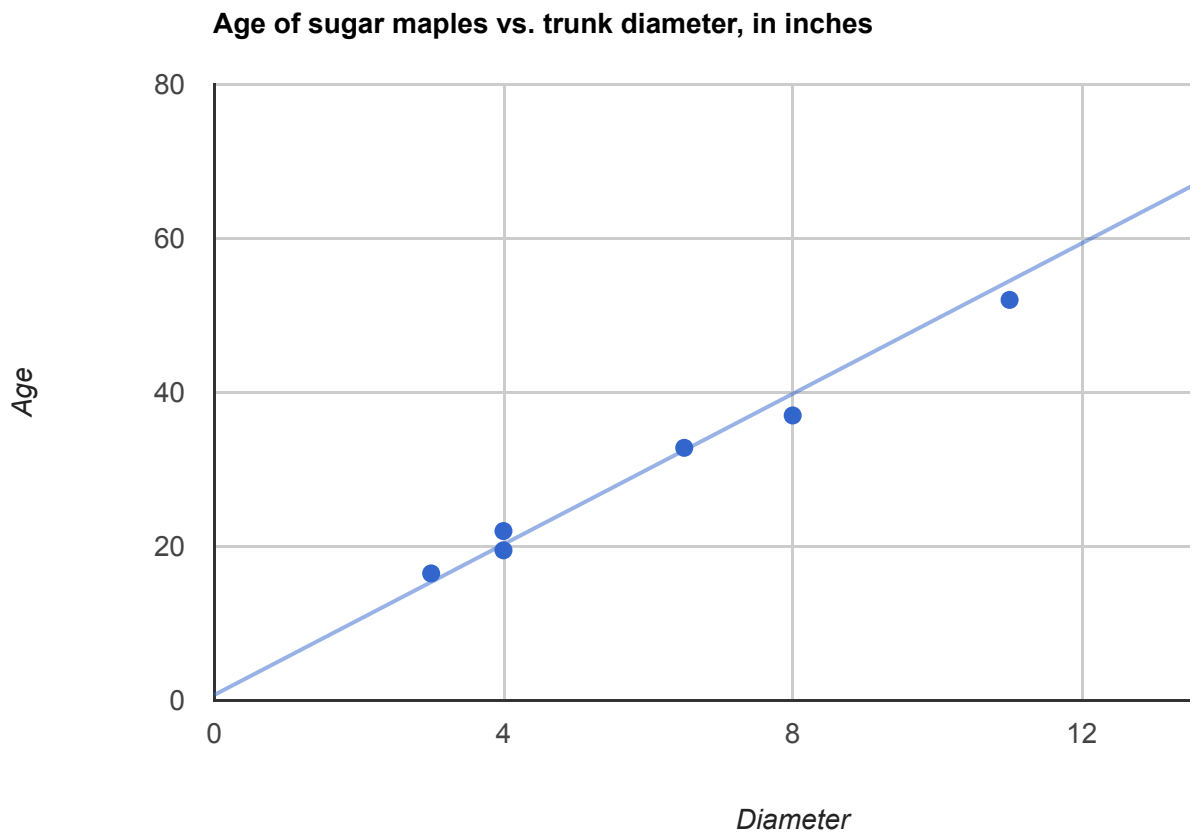
Google Charts supports three types of trendlines: linear, polynomial, and exponential.

## Linear trendlines

---

A linear trendline is the straight line that most closely approximates the data in the chart. (To be precise, it's the line that minimizes the sum of squared distances from every point to it.)

In the chart below, you can see a linear trendline on a scatter chart comparing the age of sugar maples to the diameter of their trunks. You can hover over the trendline to see the equation calculated by Google Charts:  $4.885 \text{ times the diameter} + 0.730$ .



To draw a trendline on a chart, use the `trendlines` option and specify which data series to use:

```
google.charts.setOnLoadCallback(drawChart);
function drawChart() {
  var data = google.visualization.arrayToDataTable([
    ['Diameter', 'Age'],
    [8, 37], [4, 19.5], [11, 52], [4, 22], [3, 16.5], [6.5, 32.8], [14, 72]]

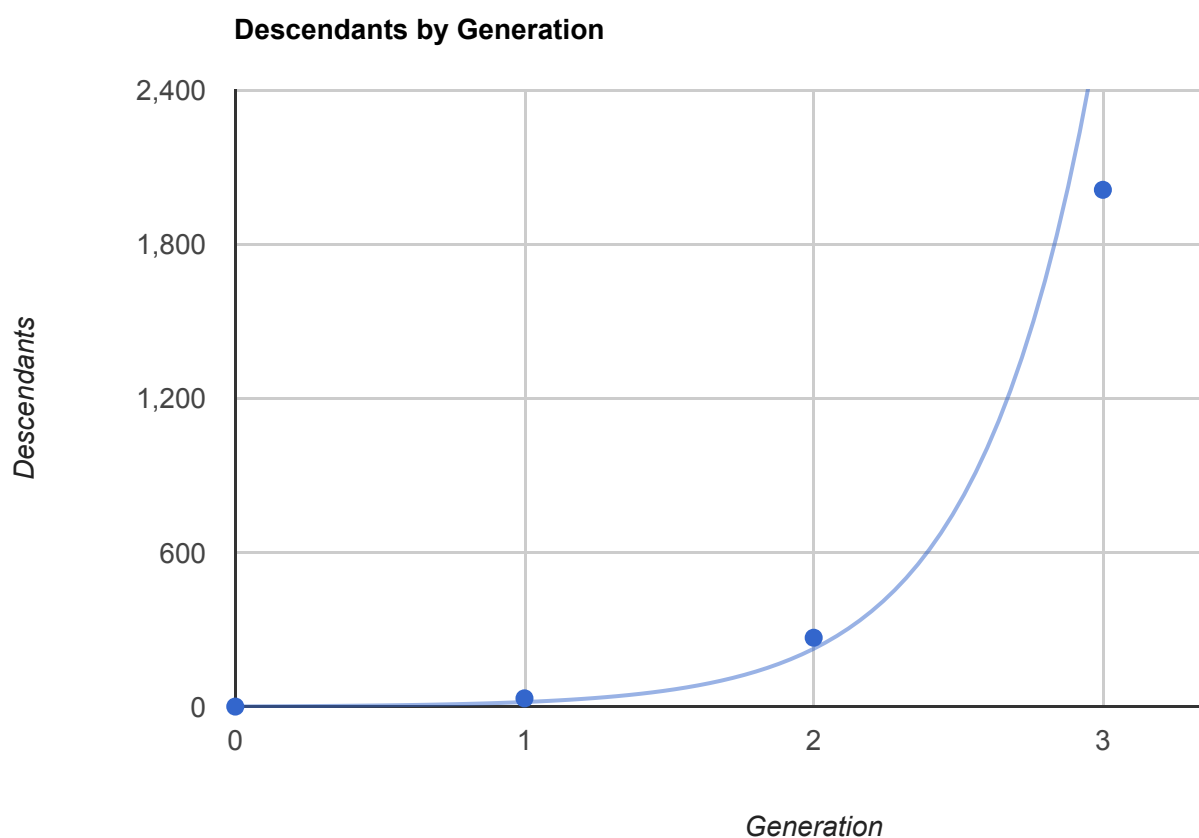
  var options = {
    title: 'Age of sugar maples vs. trunk diameter, in inches',
    hAxis: {title: 'Diameter'},
    vAxis: {title: 'Age'},
    legend: 'none',
    trendlines: { 0: {} }    // Draw a trendline for data series 0.
  };

  var chart = new google.visualization.ScatterChart(document.getElementById('chart'));
  chart.draw(data, options);
}
```

Linear trendlines are the most common type of trendline. But sometimes a curve is best for describing data, and for that, we'll need another type of trendline.

## Exponential trendlines

If your data is best explained by an exponential of the form  $e^{ax+b}$ , you can use the `type` attribute to specify an exponential trendline, as shown below:



**Note:** Unlike linear trendlines, there are several different ways to compute exponential trendlines. We provide only one method right now, but will support more in the future, and so it's possible that the name or behavior of the current exponential trendline will change.

For this chart, we also use `visibleInLegend: true` to display the exponential curve in the legend.

```
google.charts.setOnLoadCallback(drawChart);  
function drawChart() {  
  var data = google.visualization.arrayToDataTable([
```

```

    ['Generation', 'Descendants'],
    [0, 1], [1, 33], [2, 269], [3, 2013]
  ]));

  var options = {
    title: 'Descendants by Generation',
    hAxis: {title: 'Generation', minValue: 0, maxValue: 3},
    vAxis: {title: 'Descendants', minValue: 0, maxValue: 2100},
    trendlines: {
      0: {
        type: 'exponential',
        visibleInLegend: true,
      }
    }
  };

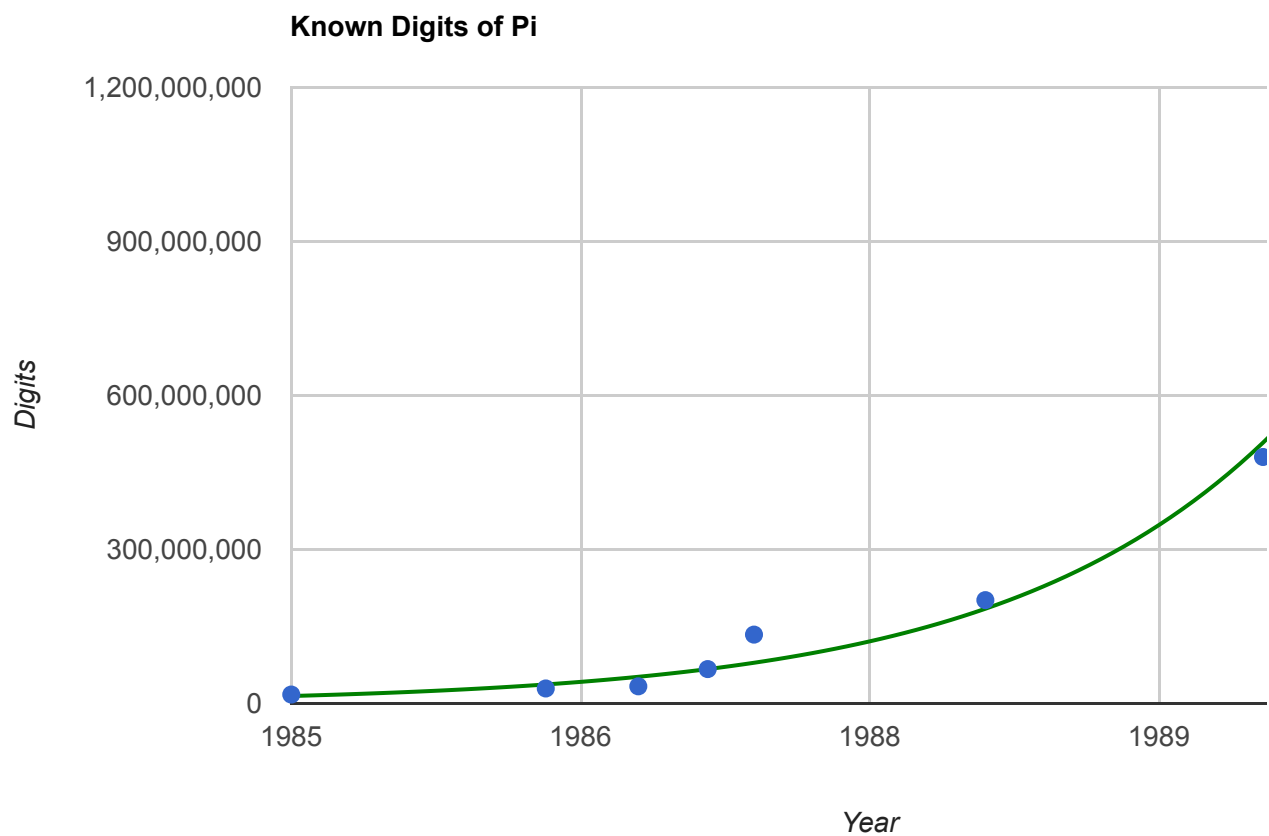
  var chart = new google.visualization.ScatterChart(document.getElementById('chart'));
  chart.draw(data, options);
}

```

## Changing the color

---

By default, trendlines are colored the same as the data series, but lighter. You can override that with the `color` attribute. Here, we chart how many digits of  $\pi$  have been calculated by year during a computationally fruitful period, coloring the exponential trendline green.



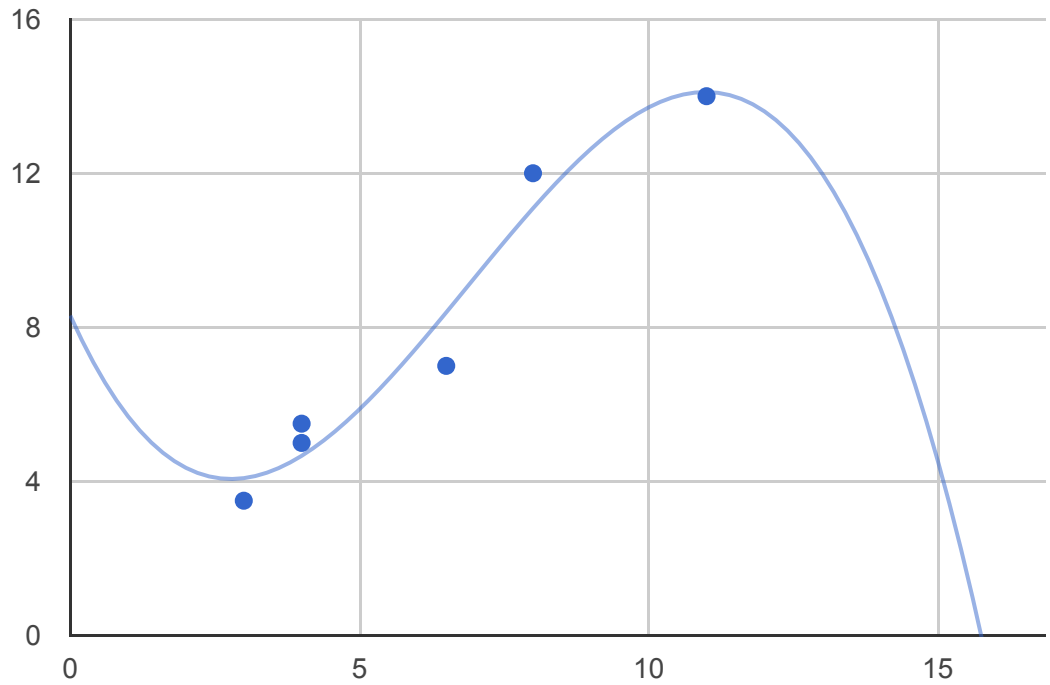
Here's the trendlines spec:

```
trendlines: {  
  0: {  
    type: 'exponential',  
    color: 'green',  
  }  
}
```

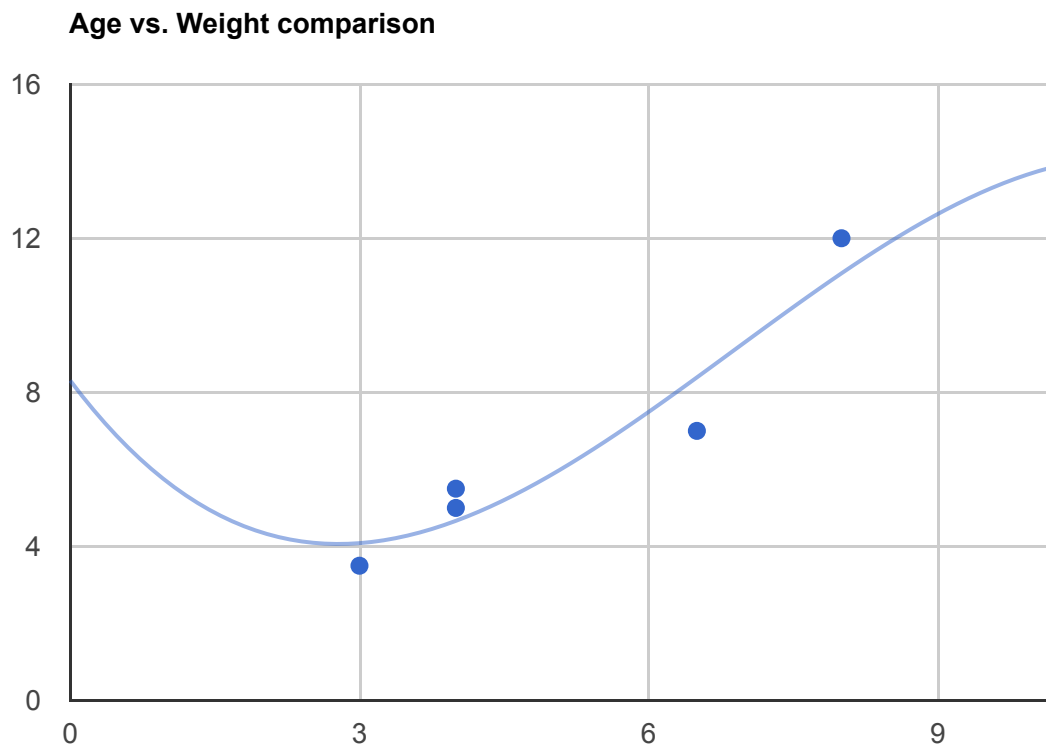
## Polynomial trendlines

To generate a polynomial trendline, specify type `polynomial` and a `degree`. Use with caution, since they can sometimes lead to misleading results. In the example below, where a roughly linear dataset is plotted with a cubic (degree 3) trendline:

**Age vs. Weight comparison**



Note that the plummet after the last data point is only visible because we artificially extended the horizontal axis to 15. Without setting `hAxis.maxValue` to 15, it would have looked like this:



Same data, same polynomial, different window on the data.

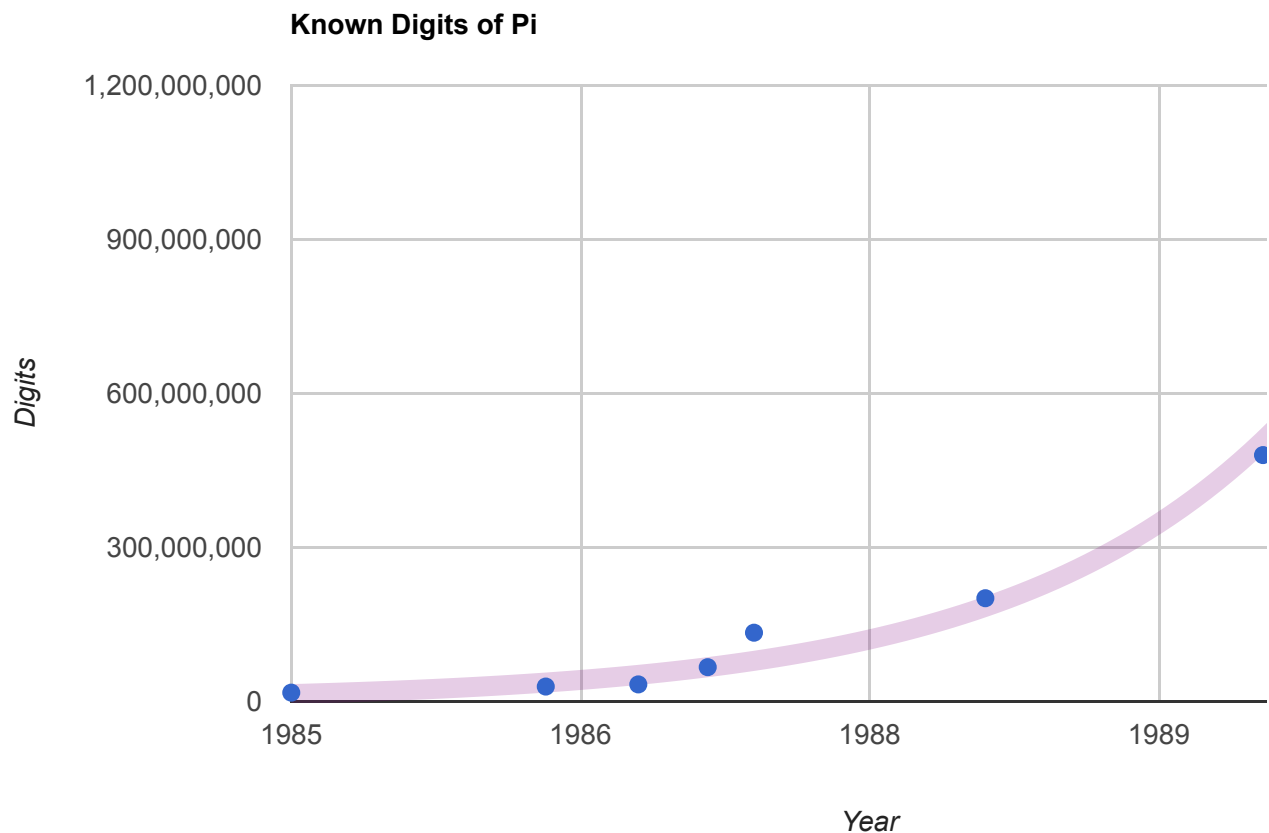
#### OPTIONS

#### FULL HTML

```
var options = {
  title: 'Age vs. Weight comparison',
  legend: 'none',
  crosshair: { trigger: "both", orientation: "both" },
  trendlines: {
    0: {
      type: 'polynomial',
      degree: 3,
      visibleInLegend: true,
    }
  }
};
```

Changing the opacity and line width

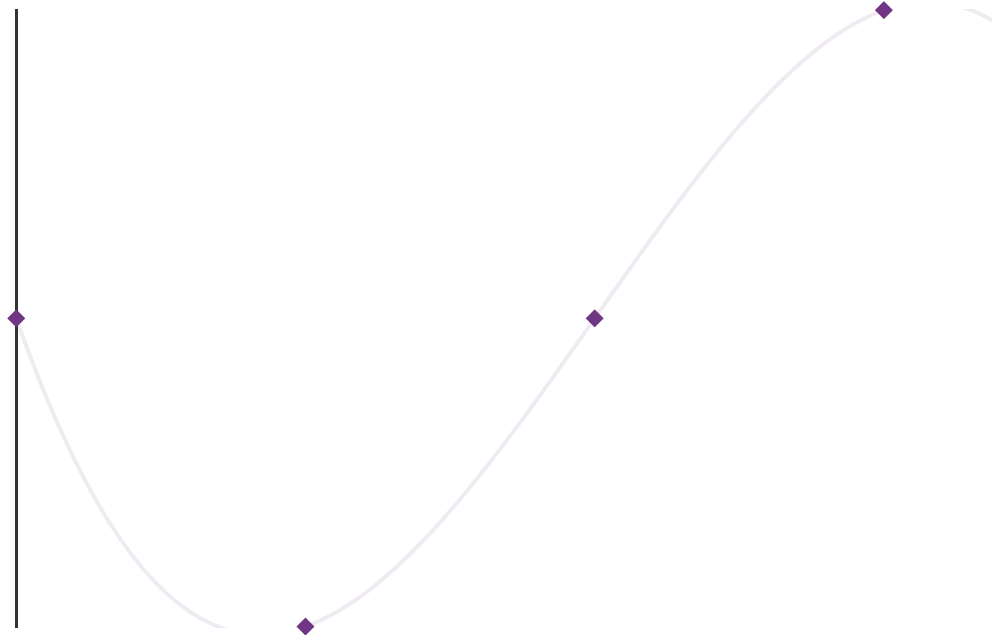
You can change the transparency of the trendline by setting `opacity` to a value between 0.0 and 1.0, and the line width by setting the `lineWidth` option.



```
trendlines: {  
  0: {  
    color: 'purple',  
    lineWidth: 10,  
    opacity: 0.2,  
    type: 'exponential'  
  }  
}
```

The `lineWidth` option will be enough for most uses, but if you like the look there's a `pointSize` option that can be used to customize the size of the selectable dots inside the trendline:





Just like light is both a wave and a particle, a trendline is both a line and a bunch of points. What users see depend on how they interact with it: normally a line, but when they hover over the trendline, a particular point will be highlighted. That point will be have a diameter equal to:

- the trendline `pointSize` if defined, else...
- the global `pointSize` if defined, else...
- 7

However, if you set the either the global or the trendline `pointSize` option, all of the selectable points will be shown, independent of the trendline's `lineWidth`.

#### OPTIONS

#### FULL HTML

```
var options = {  
  legend: 'none',  
  hAxis: { ticks: [] },  
  vAxis: { ticks: [] },  
  pointShape: 'diamond',
```

```

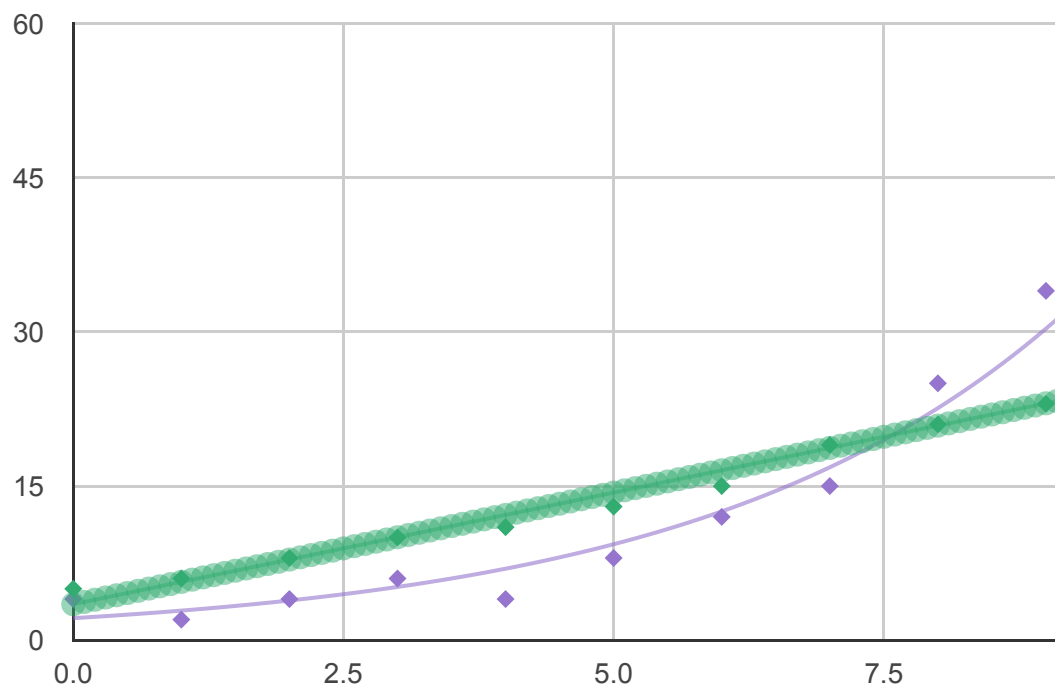
trendlines: {
  0: {
    type: 'polynomial',
    degree: 3,
    visibleInLegend: true,
    pointSize: 20, // Set the size of the trendline dots.
    opacity: 0.1
  }
}
};

```

## Making points visible

Trendlines are constructed by stamping a bunch of dots on the chart. The trendline's `pointsVisible` option determines whether the points for a particular trendline are visible. The default option for all trendlines is `true`, but if you wanted to turn off point visibility for your first trendline, set `trendlines.0.pointsVisible: false`.

The chart below demonstrates controlling the visibility of points on a per-trendline basis.



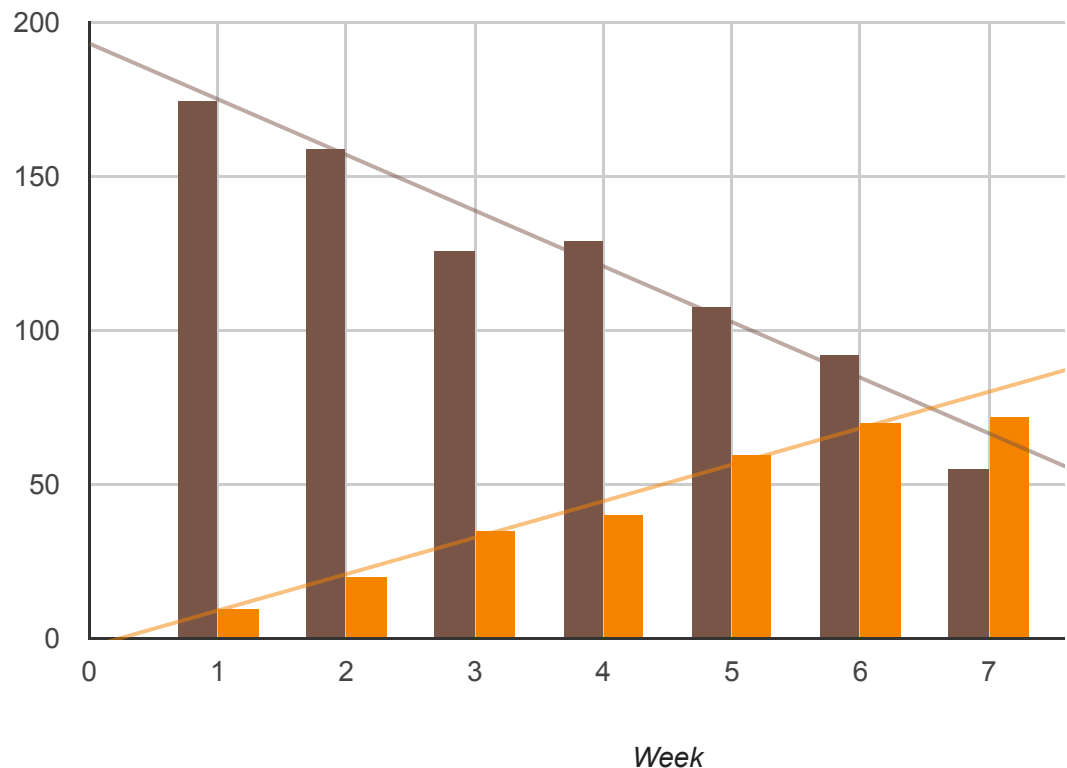
**OPTIONS****FULL HTML**

```
var options = {
  height: 500,
  legend: 'none',
  colors: ['#9575cd', '#33ac71'],
  pointShape: 'diamond',
  trendlines: {
    0: {
      type: 'exponential',
      pointSize: 20,
      opacity: 0.6,
      pointsVisible: false
    },
    1: {
      type: 'linear',
      pointSize: 10,
      pointsVisible: true
    }
  }
};
```

## Changing the label

By default, if you select `visibleInLegend`, the label reveals the equation of the trendline. You can use `labelInLegend` to specify a different label. Here, we display a trendline for each of two series, setting the labels in the legend to "Bug line" (for series 0) and "Test line" (series 1).

Google Emotion Robot: Bugs and Tests

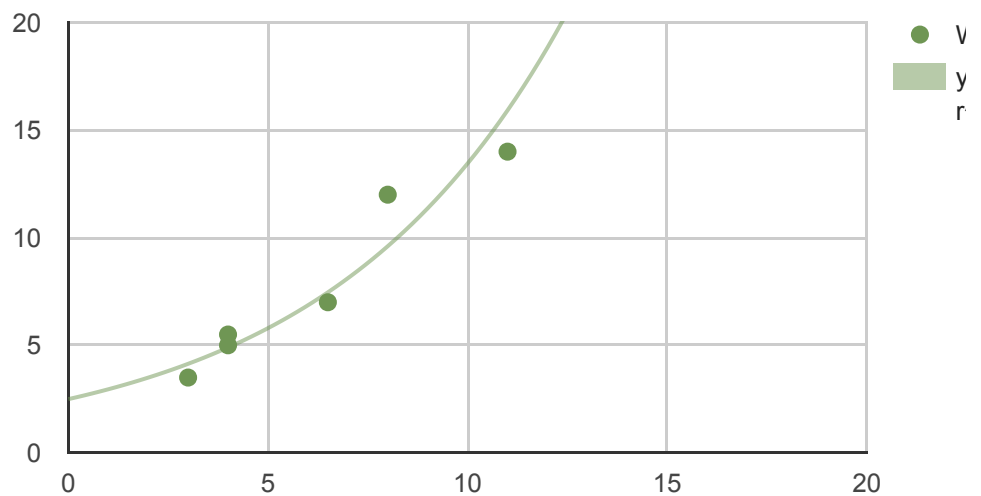
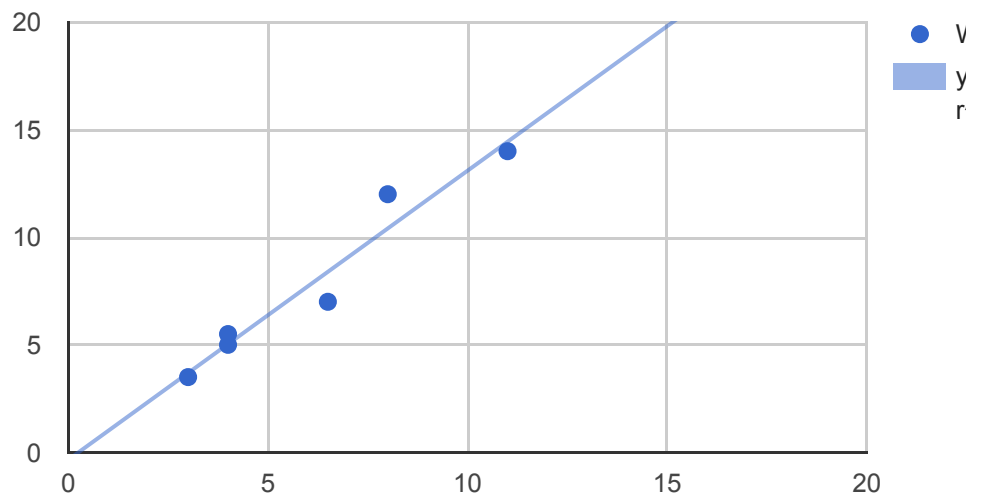


```
trendlines: {
  0: {
    labelInLegend: 'Bug line',
    visibleInLegend: true,
  },
  1: {
    labelInLegend: 'Test line',
    visibleInLegend: true,
  }
}
```

## Correlations

The [coefficient of determination](http://en.wikipedia.org/wiki/Coefficient_of_determination) ([http://en.wikipedia.org/wiki/Coefficient\\_of\\_determination](http://en.wikipedia.org/wiki/Coefficient_of_determination)), called  $R^2$  in statistics, identifies how closely a trendline matches the data. A perfect correlation is 1.0, and a perfect anticorrelation is 0.0.

You can depict  $R^2$  in the legend of your chart by setting the `showR2` option to `true`.



[CODE IT YOURSELF ON JSFIDDLE](#)

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

```

var data = google.visualization.arrayToDataTable([
  ['Age', 'Weight'],
  [ 8,    12],
  [ 4,    5.5],
  [ 11,   14],
  [ 4,    5],
  [ 3,    3.5],
  [ 6.5,  7]
]);

var options = {
  hAxis: {minValue: 0, maxValue: 15},
  vAxis: {minValue: 0, maxValue: 15},
  chartArea: {width:'50%'},
  trendlines: {
    0: {
      type: 'linear',
      showR2: true,
      visibleInLegend: true
    }
  }
};

var chartLinear = new google.visualization.ScatterChart(document.getElementById('chartLinear'));
chartLinear.draw(data, options);

options.trendlines[0].type = 'exponential';
options.colors = ['#6F9654'];

var chartExponential = new google.visualization.ScatterChart(document.getElementById('chartExponential'));
chartExponential.draw(data, options);
}
</script>
</head>
<body>
  <div id="chartLinear" style="height: 350px; width: 800px"></div>
  <div id="chartExponential" style="height: 350px; width: 800px"></div>
</body>
</html>

```

---

Except as otherwise noted, the content of this page is licensed under the [Creative Commons Attribution 3.0 License](http://creativecommons.org/licenses/by/3.0/) (<http://creativecommons.org/licenses/by/3.0/>), and code samples are licensed under the [Apache 2.0 License](http://www.apache.org/licenses/LICENSE-2.0) (<http://www.apache.org/licenses/LICENSE-2.0>). For details, see our [Site Policies](https://developers.google.com/terms/site-policies) (<https://developers.google.com/terms/site-policies>). Java is a registered trademark of Oracle and/or its affiliates.

上次更新日期: 二月 23, 2017