D3 API Reference

D3 is a collection of modules that are designed to work together; you can use the modules independently, or you can use them together as part of the default build. The source and documentation for each module is available in its repository. Follow the links below to learn more. For changes between major versions, see CHANGES; see also the release notes and the 3.x reference.

- Arrays (Statistics, Search, Iterables, Sets, Transformations, Histograms, Interning)
- Axes
- Brushes
- Chords
- Colors
- Color Schemes
- Contours
- Voronoi Diagrams
- Dispatches
- Dragging
- Delimiter-Separated Values
- Easings
- Fetches
- Forces
- Number Formats
- Geographies (Paths, Projections, Spherical Math, Spherical Shapes, Streams, Transforms)
- Hierarchies
- Interpolators
- Paths
- Polygons
- Quadtrees
- Random Numbers
- Scales (Continuous, Sequential, Diverging, Quantize, Ordinal)
- Selections (Selecting, Modifying, Data, Events, Control, Local Variables, Namespaces)
- Shapes (Arcs, Pies, Lines, Areas, Curves, Links, Symbols, Stacks)
- Time Formats
- Time Intervals
- Timers
- Transitions
- Zooming

D3 uses semantic versioning. The current version is exposed as d3.version.

Arrays (d3-array)

Array manipulation, ordering, searching, summarizing, etc.

Statistics

Methods for computing basic summary statistics.

- d3.min - compute the minimum value in an iterable.
- d3.minIndex - compute the index of the minimum value in an iterable.
- d3.max - compute the maximum value in an iterable.
- d3.maxIndex - compute the index of the maximum value in an iterable.
- d3.extent - compute the minimum and maximum value in an iterable.
- d3.sum - compute the sum of an iterable of numbers.
- d3.mean - compute the arithmetic mean of an iterable of numbers.
- d3.median - compute the median of an iterable of numbers (the 0.5-quantile).
- d3.mode - compute the mode (the most common value) of an iterable of numbers.
- d3.cumsum - compute the cumulative sum of an iterable.
- d3.rank - compute the rank order of an iterable.
- d3.quantile - compute a quantile for an iterable of numbers.
- d3.quantileSorted - compute a quantile for a sorted array of numbers.
- d3.variance - compute the variance of an iterable of numbers.
- d3.deviation - compute the standard deviation of an iterable of numbers.
- d3.fcumsum - compute a full precision cumulative summation of numbers.
- d3.fsum - compute a full precision summation of an iterable of numbers.
- new d3.Adder - creates a full precision adder.
- *adder*.add - add a value to an adder.
- *adder*.valueOf - returns a double precision representation of an adder's value.

Search

Methods for searching arrays for a specific element.

- d3.least - returns the least element of an iterable.
- d3.leastIndex - returns the index of the least element of an iterable.
- d3.greatest - returns the greatest element of an iterable.
- d3.greatestIndex - returns the index of the greatest element of an iterable.
- d3.bisectCenter - binary search for a value in a sorted array.
- d3.bisectLeft - binary search for a value in a sorted array.
- d3.bisect - binary search for a value in a sorted array.
- d3.bisectRight - binary search for a value in a sorted array.
- d3.bisector - bisect using an accessor or comparator.
- *bisector*.center - binary search for a value in a sorted array.
- *bisector*.left - bisectLeft, with the given comparator.
- *bisector*.right - bisectRight, with the given comparator.
- d3.quickselect - reorder an array of numbers.
- d3.ascending - compute the natural order of two values.
- d3.descending - compute the natural order of two values.

Transformations

Methods for transforming arrays and for generating new arrays.

- d3.flatGroup - group an iterable into a flat array.
- d3.flatRollup - reduce an iterable into a flat array.
- d3.group - group an iterable into a nested Map.
- d3.groups - group an iterable into a nested array.
- d3.groupSort - sort keys according to grouped values.
- d3.index - index an iterable into a nested Map.
- d3.indexes - index an iterable into a nested array.
- d3.rollup - reduce an iterable into a nested Map.
- d3.rollups - reduce an iterable into a nested array.
- d3.count - count valid number values in an iterable.
- d3.cross - compute the Cartesian product of two iterables.
- d3.merge - merge multiple iterables into one array.
- d3.pairs - create an array of adjacent pairs of elements.
- d3.permute - reorder an iterable of elements according to an iterable of indexes.
- d3.shuffle - randomize the order of an iterable.
- d3.shuffler - randomize the order of an iterable.
- d3.ticks - generate representative values from a numeric interval.
- d3.tickIncrement - generate representative values from a numeric interval.
- d3.tickStep - generate representative values from a numeric interval.
- d3.nice - extend an interval to align with ticks.
- d3.range - generate a range of numeric values.
- d3.transpose - transpose an array of arrays.
- d3.zip - transpose a variable number of arrays.

Iterables

- d3.every - test if all values satisfy a condition.
- d3.some - test if any value satisfies a condition.
- d3.filter - filter values.
- d3.map - map values.
- d3.reduce - reduce values.
- d3.reverse - reverse the order of values.
- d3.sort - sort values.

Sets

- d3.difference - compute a set difference.
- d3.disjoint - test whether two sets are disjoint.
- d3.intersection - compute a set intersection.
- d3.superset - test whether a set is a superset of another.
- d3.subset - test whether a set is a subset of another.
- d3.union - compute a set union.

Histograms

Bin discrete samples into continuous, non-overlapping intervals.

- d3.bin - create a new bin generator.
- *bin* - bins a given array of samples.
- *bin*.value - specify a value accessor for each sample.
- *bin*.domain - specify the interval of observable values.
- *bin*.thresholds - specify how values are divided into bins.
- d3.thresholdFreedmanDiaconis - the Freedman–Diaconis binning rule.
- d3.thresholdScott - Scott's normal reference binning rule.
- d3.thresholdSturges - Sturges' binning formula.

Interning

- d3.InternMap - a key-interning Map.
- d3.InternSet - a value-interning Set.

Axes (d3-axis)

Human-readable reference marks for scales.

- d3.axisTop - create a new top-oriented axis generator.
- d3.axisRight - create a new right-oriented axis generator.
- d3.axisBottom - create a new bottom-oriented axis generator.
- d3.axisLeft - create a new left-oriented axis generator.
- *axis* - generate an axis for the given selection.
- *axis*.scale - set the scale.
- *axis*.ticks - customize how ticks are generated and formatted.
- *axis*.tickArguments - customize how ticks are generated and formatted.
- *axis*.tickValues - set the tick values explicitly.
- *axis*.tickFormat - set the tick format explicitly.
- *axis*.tickSize - set the size of the ticks.
- *axis*.tickSizeInner - set the size of inner ticks.
- *axis*.tickSizeOuter - set the size of outer (extent) ticks.
- *axis*.tickPadding - set the padding between ticks and labels.
- *axis*.offset - set the pixel offset for crisp edges.

Brushes (d3-brush)

Select a one- or two-dimensional region using the mouse or touch.

- d3.brush - create a new two-dimensional brush.
- d3.brushX - create a brush along the x-dimension.
- d3.brushY - create a brush along the y-dimension.
- *brush* - apply the brush to a selection.
- *brush*.move - move the brush selection.
- *brush*.clear - clear the brush selection.

- *brush*.extent - define the brushable region.
- *brush*.filter - control which input events initiate brushing.
- *brush*.touchable - set the touch support detector.
- *brush*.keyModifiers - enable or disable key interaction.
- *brush*.handleSize - set the size of the brush handles.
- *brush*.on - listen for brush events.
- d3.brushSelection - get the brush selection for a given node.

Chords (d3-chord)

- d3.chord - create a new chord layout.
- *chord* - compute the layout for the given matrix.
- *chord*.padAngle - set the padding between adjacent groups.
- *chord*.sortGroups - define the group order.
- *chord*.sortSubgroups - define the source and target order within groups.
- *chord*.sortChords - define the chord order across groups.
- d3.chordDirected - create a directed chord generator.
- d3.chordTranspose - create a transposed chord generator.
- d3.ribbon - create a ribbon shape generator.
- *ribbon* - generate a ribbon shape.
- *ribbon*.source - set the source accessor.
- *ribbon*.target - set the target accessor.
- *ribbon*.radius - set the ribbon source and target radius.
- *ribbon*.sourceRadius - set the ribbon source radius.
- *ribbon*.targetRadius - set the ribbon target radius.
- *ribbon*.startAngle - set the ribbon source or target start angle.
- *ribbon*.endAngle - set the ribbon source or target end angle.
- *ribbon*.padAngle - set the pad angle accessor.
- *ribbon*.context - set the render context.
- d3.ribbonArrow - create an arrow ribbon generator.
- *ribbonArrow*.headRadius - set the arrowhead radius accessor.

Colors (d3-color)

Color manipulation and color space conversion.

- d3.color - parse the given CSS color specifier.
- *color*.opacity - the color's opacity.
- *color*.rgb - compute the RGB equivalent of this color.
- *color*.copy - return a copy of this color.
- *color*.brighter - create a brighter copy of this color.
- *color*.darker - create a darker copy of this color.
- *color*.displayable - returns true if the color is displayable on standard hardware.
- *color*.formatHex - returns the hexadecimal RRGGBB string representation of this color.

- *color*.formatHex8 - returns the hexadecimal RRGGBBAA string representation of this color.
- *color*.formatHsl - returns the RGB string representation of this color.
- *color*.formatRgb - returns the HSL string representation of this color.
- *color*.toString - returns the RGB string representation of this color.
- d3.rgb - create a new RGB color.
- *rgb*.clamp - returns copy of this color clamped to the RGB color space.
- d3.hsl - create a new HSL color.
- *hsl*.clamp - returns copy of this color clamped to the HSL color space.
- d3.lab - create a new Lab color.
- d3.gray - create a new Lab gray.
- d3.hcl - create a new HCL color.
- d3.lch - create a new HCL color.
- d3.cubehelix - create a new Cubehelix color.

Color Schemes (d3-scale-chromatic)

Color ramps and palettes for quantitative, ordinal and categorical scales.

Categorical

- d3.schemeCategory10 - an array of ten categorical colors.
- d3.schemeAccent - an array of eight categorical colors.
- d3.schemeDark2 - an array of eight categorical colors.
- d3.schemePaired - an array of twelve categorical colors.
- d3.schemePastel1 - an array of nine categorical colors.
- d3.schemePastel2 - an array of eight categorical colors.
- d3.schemeSet1 - an array of nine categorical colors.
- d3.schemeSet2 - an array of eight categorical colors.
- d3.schemeSet3 - an array of twelve categorical colors.
- d3.schemeTableau10 - an array of ten categorical colors.

Diverging

- d3.interpolateBrBG - ColorBrewer BrBG interpolator.
- d3.interpolatePiYG - ColorBrewer PiYG interpolator.
- d3.interpolatePRGn - ColorBrewer PRGn interpolator.
- d3.interpolatePuOr - ColorBrewer PuOr interpolator.
- d3.interpolateRdBu - ColorBrewer RdBu interpolator.
- d3.interpolateRdGy - ColorBrewer RdGy interpolator.
- d3.interpolateRdYlBu - ColorBrewer RdYlBu interpolator.
- d3.interpolateRdYlGn - ColorBrewer RdYlGn interpolator.
- d3.interpolateSpectral - ColorBrewer spectral interpolator.
- d3.schemeBrBG - ColorBrewer BrBG scheme.
- d3.schemePiYG - ColorBrewer PiYG scheme.
- d3.schemePRGn - ColorBrewer PRGn scheme.

- d3.schemePuOr - ColorBrewer PuOr scheme.
- d3.schemeRdBu - ColorBrewer RdBu scheme.
- d3.schemeRdGy - ColorBrewer RdGy scheme.
- d3.schemeRdYlBu - ColorBrewer RdYlBu scheme.
- d3.schemeRdYlGn - ColorBrewer RdYlGn scheme.
- d3.schemeSpectral - ColorBrewer spectral scheme.

Sequential (Single Hue)

- d3.interpolateBlues -
- d3.interpolateGreens -
- d3.interpolateGreys -
- d3.interpolateOranges -
- d3.interpolatePurples -
- d3.interpolateReds -
- d3.schemeBlues -
- d3.schemeGreens -
- d3.schemeGreys -
- d3.schemeOranges -
- d3.schemePurples -
- d3.schemeReds -

Sequential (Multi-Hue)

- d3.interpolateBuGn - ColorBrewer BuGn interpolator.
- d3.interpolateBuPu - ColorBrewer BuPu interpolator.
- d3.interpolateCividis - cividis interpolator.
- d3.interpolateCool - cool interpolator.
- d3.interpolateCubehelixDefault - cubehelix interpolator.
- d3.interpolateGnBu - ColorBrewer GnBu interpolator.
- d3.interpolateInferno - inferno interpolator.
- d3.interpolateMagma - magma interpolator.
- d3.interpolateOrRd - ColorBrewer OrRd interpolator.
- d3.interpolatePlasma - plasma interpolator.
- d3.interpolatePuBu - ColorBrewer PuBu interpolator.
- d3.interpolatePuBuGn - ColorBrewer PuBuGn interpolator.
- d3.interpolatePuRd - ColorBrewer PuRd interpolator.
- d3.interpolateRdPu - ColorBrewer RdPu interpolator.
- d3.interpolateTurbo - turbo interpolator.
- d3.interpolateViridis - viridis interpolator.
- d3.interpolateWarm - warm interpolator.
- d3.interpolateYlGn - ColorBrewer YlGn interpolator.
- d3.interpolateYlGnBu - ColorBrewer YlGnBu interpolator.
- d3.interpolateYlOrBr - ColorBrewer YlOrBr interpolator.
- d3.interpolateYlOrRd - ColorBrewer YlOrRd interpolator.
- d3.schemeBuGn - ColorBrewer BuGn scheme.

- d3.schemeBuPu - ColorBrewer BuPu scheme.
- d3.schemeGnBu - ColorBrewer GnBu scheme.
- d3.schemeOrRd - ColorBrewer OrRd scheme.
- d3.schemePuBu - ColorBrewer PuBu scheme.
- d3.schemePuBuGn - ColorBrewer PuBuGn scheme.
- d3.schemePuRd - ColorBrewer PuRd scheme.
- d3.schemeRdPu - ColorBrewer RdPu scheme.
- d3.schemeYlGn - ColorBrewer YlGn scheme.
- d3.schemeYlGnBu - ColorBrewer YlGnBu scheme.
- d3.schemeYlOrBr - ColorBrewer YlOrBr scheme.
- d3.schemeYlOrRd - ColorBrewer YlOrRd scheme.

Cyclical

- d3.interpolateRainbow - the “less-angry” rainbow
- d3.interpolateSinebow - the “sinebow” smooth rainbow

Contours (d3-contour)

Compute contour polygons using marching squares.

- d3.contours - create a new contour generator.
- *contours* - compute the contours for a given grid of values.
- *contours*.contour - compute a contour for a given value.
- *contours*.size - set the size of a contour generator.
- *contours*.smooth - set whether or not the generated contours are smoothed.
- *contours*.thresholds - set the thresholds of a contour generator.
- d3.contourDensity - create a new density estimator.
- *density* - estimate the density of a given array of samples.
- *density*.x - set the x accessor of the density estimator.
- *density*.y - set the y accessor of the density estimator.
- *density*.weight - set the *weight* accessor of the density estimator.
- *density*.size - set the size of the density estimator.
- *density*.cellSize - set the cell size of the density estimator.
- *density*.thresholds - set the thresholds of the density estimator.
- *density*.bandwidth - set the bandwidth of the density estimator.

Voronoi Diagrams (d3-delaunay)

Compute the Voronoi diagram of a set of two-dimensional points.

- new Delaunay - create a delaunay triangulation for an array of point coordinates.
- Delaunay.from - create a delaunay triangulation for an iterable of points.
- *delaunay*.points - the coordinates of the points.
- *delaunay*.halfedges - the delaunay halfedges.
- *delaunay*.hull - the convex hull as point indices.

- *delaunay*.triangles - the delaunay triangles.
- *delaunay*.inedges - the delaunay inedges
- *delaunay*.find - find the closest point in the delaunay triangulation.
- *delaunay*.neighbors - the neighbors of a point in the delaunay triangulation.
- *delaunay*.render - render the edges of the delaunay triangulation.
- *delaunay*.renderHull - render the convex hull.
- *delaunay*.renderTriangle - render a triangle.
- *delaunay*.renderPoints - render the points.
- *delaunay*.hullPolygon - the closed convex hull as point coordinates.
- *delaunay*.trianglePolygons - iterate over all the triangles as polygons.
- *delaunay*.trianglePolygon - return a triangle as a polygon.
- *delaunay*.update - update a delaunay triangulation in place.
- *delaunay*.voronoi - compute the voronoi diagram associated with a delaunay triangulation.
- *voronoi*.delaunay - the voronoi diagram's source delaunay triangulation.
- *voronoi*.circumcenters - the triangles' circumcenters.
- *voronoi*.vectors - directions for the outer (infinite) cells of the voronoi diagram.
- *voronoi*.xmin - set the *xmin* bound of the extent.
- *voronoi*.ymin - set the *ymin* bound of the extent.
- *voronoi*.xmax - set the *xmax* bound of the extent.
- *voronoi*.ymax - set the *ymax* bound of the extent.
- *voronoi*.contains - test whether a point is inside a voronoi cell.
- *voronoi*.neighbors - the neighbors of a point in the voronoi diagram.
- *voronoi*.render - render the mesh of voronoi cells.
- *voronoi*.renderBounds - render the extent.
- *voronoi*.renderCell - render a voronoi cell.
- *voronoi*.cellPolygons - iterate over all the cells as polygons.
- *voronoi*.cellPolygon - return a cell as a polygon.
- *voronoi*.update - update a voronoi diagram in place.

Dispatches (d3-dispatch)

Separate concerns using named callbacks.

- d3.dispatch - create a custom event dispatcher.
- *dispatch*.on - register or unregister an event listener.
- *dispatch*.copy - create a copy of a dispatcher.
- *dispatch*.call - dispatch an event to registered listeners.
- *dispatch*.apply - dispatch an event to registered listeners.

Dragging (d3-drag)

Drag and drop SVG, HTML or Canvas using mouse or touch input.

- d3.drag - create a drag behavior.
- *drag* - apply the drag behavior to a selection.

- *drag*.container - set the coordinate system.
- *drag*.filter - ignore some initiating input events.
- *drag*.touchable - set the touch support detector.
- *drag*.subject - set the thing being dragged.
- *drag*.clickDistance - set the click distance threshold.
- *drag*.on - listen for drag events.
- d3.dragDisable - prevent native drag-and-drop and text selection.
- d3.dragEnable - enable native drag-and-drop and text selection.
- *event*.on - listen for drag events on the current gesture.

Delimiter-Separated Values (d3-dsv)

Parse and format delimiter-separated values, most commonly CSV and TSV.

- d3.csvParse - parse the given CSV string, returning an array of objects.
- d3.csvParseRows - parse the given CSV string, returning an array of rows.
- d3.csvFormat - format the given array of objects as CSV.
- d3.csvFormatBody - format the given array of objects as CSV.
- d3.csvFormatRows - format the given array of rows as CSV.
- d3.csvFormatRow - format the given row as CSV.
- d3.csvFormatValue - format the given value as CSV.
- d3.tsvParse - parse the given TSV string, returning an array of objects.
- d3.tsvParseRows - parse the given TSV string, returning an array of rows.
- d3.tsvFormat - format the given array of objects as TSV.
- d3.tsvFormatBody - format the given array of objects as TSV.
- d3.tsvFormatRows - format the given array of rows as TSV.
- d3.tsvFormatRow - format the given row as TSV.
- d3.tsvFormatValue - format the given value as TSV.
- d3.dsvFormat - create a new parser and formatter for the given delimiter.
- *dsv*.parse - parse the given string, returning an array of objects.
- *dsv*.parseRows - parse the given string, returning an array of rows.
- *dsv*.format - format the given array of objects.
- *dsv*.formatBody - format the given array of objects.
- *dsv*.formatRows - format the given array of rows.
- *dsv*.formatRow - format the given row.
- *dsv*.formatValue - format the given value.
- d3.autoType - automatically infer value types for the given object.

Easings (d3-ease)

Easing functions for smooth animation.

- *ease* - ease the given normalized time.
- d3.easeLinear - linear easing; the identity function.
- d3.easePolyIn - polynomial easing; raises time to the given power.
- d3.easePolyOut - reverse polynomial easing.
- d3.easePoly - an alias for easePolyInOut.

- d3.easePolyInOut - symmetric polynomial easing.
- *poly*.exponent - specify the polynomial exponent.
- d3.easeQuadIn - quadratic easing; squares time.
- d3.easeQuadOut - reverse quadratic easing.
- d3.easeQuad - an alias for easeQuadInOut.
- d3.easeQuadInOut - symmetric quadratic easing.
- d3.easeCubicIn - cubic easing; cubes time.
- d3.easeCubicOut - reverse cubic easing.
- d3.easeCubic - an alias for easeCubicInOut.
- d3.easeCubicInOut - symmetric cubic easing.
- d3.easeSinIn - sinusoidal easing.
- d3.easeSinOut - reverse sinusoidal easing.
- d3.easeSin - an alias for easeSinInOut.
- d3.easeSinInOut - symmetric sinusoidal easing.
- d3.easeExpIn - exponential easing.
- d3.easeExpOut - reverse exponential easing.
- d3.easeExp - an alias for easeExpInOut.
- d3.easeExpInOut - symmetric exponential easing.
- d3.easeCircleIn - circular easing.
- d3.easeCircleOut - reverse circular easing.
- d3.easeCircle - an alias for easeCircleInOut.
- d3.easeCircleInOut - symmetric circular easing.
- d3.easeElasticIn - elastic easing, like a rubber band.
- d3.easeElastic - an alias for easeElasticOut.
- d3.easeElasticOut - reverse elastic easing.
- d3.easeElasticInOut - symmetric elastic easing.
- *elastic*.amplitude - specify the elastic amplitude.
- *elastic*.period - specify the elastic period.
- d3.easeBackIn - anticipatory easing, like a dancer bending his knees before jumping.
- d3.easeBackOut - reverse anticipatory easing.
- d3.easeBack - an alias for easeBackInOut.
- d3.easeBackInOut - symmetric anticipatory easing.
- *back*.overshoot - specify the amount of overshoot.
- d3.easeBounceIn - bounce easing, like a rubber ball.
- d3.easeBounce - an alias for easeBounceOut.
- d3.easeBounceOut - reverse bounce easing.
- d3.easeBounceInOut - symmetric bounce easing.

Fetches (d3-fetch)

Convenience methods on top of the Fetch API.

- d3.blob - get a file as a blob.
- d3.buffer - get a file as an array buffer.
- d3.csv - get a comma-separated values (CSV) file.

- d3.dsv - get a delimiter-separated values (CSV) file.
- d3.html - get an HTML file.
- d3.image - get an image.
- d3.json - get a JSON file.
- d3.svg - get an SVG file.
- d3.text - get a plain text file.
- d3.tsv - get a tab-separated values (TSV) file.
- d3.xml - get an XML file.

Forces (d3-force)

Force-directed graph layout using velocity Verlet integration.

- d3.forceSimulation - create a new force simulation.
- *simulation*.restart - reheat and restart the simulation's timer.
- *simulation*.stop - stop the simulation's timer.
- *simulation*.tick - advance the simulation one step.
- *simulation*.nodes - set the simulation's nodes.
- *simulation*.alpha - set the current alpha.
- *simulation*.alphaMin - set the minimum alpha threshold.
- *simulation*.alphaDecay - set the alpha exponential decay rate.
- *simulation*.alphaTarget - set the target alpha.
- *simulation*.velocityDecay - set the velocity decay rate.
- *simulation*.force - add or remove a force.
- *simulation*.find - find the closest node to the given position.
- *simulation*.randomSource - set the simulation's random source.
- *simulation*.on - add or remove an event listener.
- *force* - apply the force.
- *force*.initialize - initialize the force with the given nodes.
- d3.forceCenter - create a centering force.
- *center*.x - set the center x-coordinate.
- *center*.y - set the center y-coordinate.
- *center*.strength - set the strength of the centering force.
- d3.forceCollide - create a circle collision force.
- *collide*.radius - set the circle radius.
- *collide*.strength - set the collision resolution strength.
- *collide*.iterations - set the number of iterations.
- d3.forceLink - create a link force.
- *link*.links - set the array of links.
- *link*.id - link nodes by numeric index or string identifier.
- *link*.distance - set the link distance.
- *link*.strength - set the link strength.
- *link*.iterations - set the number of iterations.
- d3.forceManyBody - create a many-body force.
- *manyBody*.strength - set the force strength.
- *manyBody*.theta - set the Barnes–Hut approximation accuracy.

- *manyBody*.distanceMin - limit the force when nodes are close.
- *manyBody*.distanceMax - limit the force when nodes are far.
- d3.forceX - create an x-positioning force.
- *x*.strength - set the force strength.
- *x*.x - set the target x-coordinate.
- d3.forceY - create an y-positioning force.
- *y*.strength - set the force strength.
- *y*.y - set the target y-coordinate.
- d3.forceRadial - create a radial positioning force.
- *radial*.strength - set the force strength.
- *radial*.radius - set the target radius.
- *radial*.x - set the target center x-coordinate.
- *radial*.y - set the target center y-coordinate.

Number Formats (d3-format)

Format numbers for human consumption.

- d3.format - alias for *locale*.format on the default locale.
- d3.formatPrefix - alias for *locale*.formatPrefix on the default locale.
- *locale*.format - create a number format.
- *locale*.formatPrefix - create a SI-prefix number format.
- d3.formatSpecifier - parse a number format specifier.
- new d3.FormatSpecifier - augments a number format specifier object.
- d3.precisionFixed - compute decimal precision for fixed-point notation.
- d3.precisionPrefix - compute decimal precision for SI-prefix notation.
- d3.precisionRound - compute significant digits for rounded notation.
- d3.formatLocale - define a custom locale.
- d3.formatDefaultLocale - define the default locale.

Geographies (d3-geo)

Geographic projections, shapes and math.

Paths

- d3.geoPath - create a new geographic path generator.
- *path* - project and render the specified feature.
- *path*.area - compute the projected planar area of a given feature.
- *path*.bounds - compute the projected planar bounding box of a given feature.
- *path*.centroid - compute the projected planar centroid of a given feature.
- *path*.measure - compute the projected planar length of a given feature.
- *path*.projection - set the geographic projection.
- *path*.context - set the render context.
- *path*.pointRadius - set the radius to display point features.

Projections

- *projection* - project the specified point from the sphere to the plane.
- *projection*.invert - unproject the specified point from the plane to the sphere.
- *projection*.stream - wrap the specified stream to project geometry.
- *projection*.preclip - set the projection's spherical clipping function.
- *projection*.postclip - set the projection's cartesian clipping function.
- *projection*.clipAngle - set the radius of the clip circle.
- *projection*.clipExtent - set the viewport clip extent, in pixels.
- *projection*.scale - set the scale factor.
- *projection*.translate - set the translation offset.
- *projection*.center - set the center point.
- *projection*.angle - set the post-projection rotation.
- *projection*.reflectX - reflect the x-dimension.
- *projection*.reflectY - reflect the y-dimension.
- *projection*.rotate - set the three-axis spherical rotation angles.
- *projection*.precision - set the precision threshold for adaptive sampling.
- *projection*.fitExtent - set the scale and translate to fit a GeoJSON object.
- *projection*.fitSize - set the scale and translate to fit a GeoJSON object.
- *projection*.fitWidth - set the scale and translate to fit a GeoJSON object.
- *projection*.fitHeight - set the scale and translate to fit a GeoJSON object.
- d3.geoAzimuthalEqualArea - the azimuthal equal-area projection.
- d3.geoAzimuthalEqualAreaRaw - the raw azimuthal equal-area projection.
- d3.geoAzimuthalEquidistant - the azimuthal equidistant projection.
- d3.geoAzimuthalEquidistantRaw - the raw azimuthal equidistant projection.
- d3.geoGnomonic - the gnomonic projection.
- d3.geoGnomonicRaw - the raw gnomonic projection.
- d3.geoOrthographic - the azimuthal orthographic projection.
- d3.geoOrthographicRaw - the raw azimuthal orthographic projection.
- d3.geoStereographic - the azimuthal stereographic projection.
- d3.geoStereographicRaw - the raw azimuthal stereographic projection.
- d3.geoEqualEarth - the Equal Earth projection.
- d3.geoEqualEarthRaw - the raw Equal Earth projection.
- d3.geoAlbersUsa - a composite Albers projection for the United States.
- *conic*.parallels - set the two standard parallels.
- d3.geoAlbers - the Albers equal-area conic projection.
- d3.geoConicConformal - the conic conformal projection.
- d3.geoConicConformalRaw - the raw conic conformal projection.
- d3.geoConicEqualArea - the conic equal-area (Albers) projection.
- d3.geoConicEqualAreaRaw - the raw conic equal-area (Albers) projection.
- d3.geoConicEquidistant - the conic equidistant projection.
- d3.geoConicEquidistantRaw - the raw conic equidistant projection.
- d3.geoEquirectangular - the equirectangular (plate carreé) projection.
- d3.geoEquirectangularRaw - the raw equirectangular (plate carreé) projec-

tion.
- d3.geoMercator - the spherical Mercator projection.
- d3.geoMercatorRaw - the raw Mercator projection.
- d3.geoTransverseMercator - the transverse spherical Mercator projection.
- d3.geoTransverseMercatorRaw - the raw transverse spherical Mercator projection.
- d3.geoNaturalEarth1 - the Equal Earth projection, version 1.
- d3.geoNaturalEarth1Raw - the raw Equal Earth projection, version 1

Raw projections

- *project* - project the specified point from the sphere to the plane.
- *project*.invert - unproject the specified point from the plane to the sphere.
- d3.geoProjection - create a custom projection.
- d3.geoProjectionMutator - create a custom configurable projection.

Spherical Math

- d3.geoArea - compute the spherical area of a given feature.
- d3.geoBounds - compute the latitude-longitude bounding box for a given feature.
- d3.geoCentroid - compute the spherical centroid of a given feature.
- d3.geoDistance - compute the great-arc distance between two points.
- d3.geoLength - compute the length of a line string or the perimeter of a polygon.
- d3.geoInterpolate - interpolate between two points along a great arc.
- d3.geoContains - test whether a point is inside a given feature.
- d3.geoRotation - create a rotation function for the specified angles.
- *rotation* - rotate the given point around the sphere.
- *rotation*.invert - unrotate the given point around the sphere.

Spherical Shapes

- d3.geoCircle - create a circle generator.
- *circle* - generate a piecewise circle as a Polygon.
- *circle*.center - specify the circle center in latitude and longitude.
- *circle*.radius - specify the angular radius in degrees.
- *circle*.precision - specify the precision of the piecewise circle.
- d3.geoGraticule - create a graticule generator.
- *graticule* - generate a MultiLineString of meridians and parallels.
- *graticule*.lines - generate an array of LineStrings of meridians and parallels.
- *graticule*.outline - generate a Polygon of the graticule's extent.
- *graticule*.extent - get or set the major & minor extents.
- *graticule*.extentMajor - get or set the major extent.
- *graticule*.extentMinor - get or set the minor extent.
- *graticule*.step - get or set the major & minor step intervals.
- *graticule*.stepMajor - get or set the major step intervals.

- *graticule*.stepMinor - get or set the minor step intervals.
- *graticule*.precision - get or set the latitudinal precision.
- d3.geoGraticule10 - generate the default 10° global graticule.

Streams

- d3.geoStream - convert a GeoJSON object to a geometry stream.
- *stream*.point - indicates a point with the specified coordinates.
- *stream*.lineStart - indicates the start of a line or ring.
- *stream*.lineEnd - indicates the end of a line or ring.
- *stream*.polygonStart - indicates the start of a polygon.
- *stream*.polygonEnd - indicates the end of a polygon.
- *stream*.sphere - indicates the sphere.

Transforms

- d3.geoTransform - define a custom geometry transform.
- d3.geoIdentity - scale, translate or clip planar geometry.

Clipping

- *preclip* - pre-clipping in geographic coordinates.
- *postclip* - post-clipping in planar coordinates.
- d3.geoClipAntimeridian - cuts spherical geometries that cross the antimeridian.
- d3.geoClipCircle - clips spherical geometries to a small circle.
- d3.geoClipRectangle - clips planar geometries to a rectangular viewport.

Hierarchies (d3-hierarchy)

Layout algorithms for visualizing hierarchical data.

- d3.hierarchy - constructs a root node from hierarchical data.
- *node*.ancestors - generate an array of ancestors.
- *node*.descendants - generate an array of descendants.
- *node*.leaves - generate an array of leaves.
- *node*.find - find a node in the hierarchy.
- *node*.path - generate the shortest path to another node.
- *node*.links - generate an array of links.
- *node*.sum - evaluate and aggregate quantitative values.
- *node*.count - count the number of leaves.
- *node*.sort - sort all descendant siblings.
- *node*[Symbol.iterator] - iterate on a hierarchy.
- *node*.each - breadth-first traversal.
- *node*.eachAfter - post-order traversal.
- *node*.eachBefore - pre-order traversal.
- *node*.copy - copy a hierarchy.

- d3.stratify - create a new stratify operator.
- *stratify* - construct a root node from tabular data.
- *stratify*.id - set the node id accessor.
- *stratify*.parentId - set the parent node id accessor.
- *stratify*.path - set the path accessor.
- d3.cluster - create a new cluster (dendrogram) layout.
- *cluster* - layout the specified hierarchy in a dendrogram.
- *cluster*.size - set the layout size.
- *cluster*.nodeSize - set the node size.
- *cluster*.separation - set the separation between leaves.
- d3.tree - create a new tidy tree layout.
- *tree* - layout the specified hierarchy in a tidy tree.
- *tree*.size - set the layout size.
- *tree*.nodeSize - set the node size.
- *tree*.separation - set the separation between nodes.
- d3.treemap - create a new treemap layout.
- *treemap* - layout the specified hierarchy as a treemap.
- *treemap*.tile - set the tiling method.
- *treemap*.size - set the layout size.
- *treemap*.round - set whether the output coordinates are rounded.
- *treemap*.padding - set the padding.
- *treemap*.paddingInner - set the padding between siblings.
- *treemap*.paddingOuter - set the padding between parent and children.
- *treemap*.paddingTop - set the padding between the parent's top edge and children.
- *treemap*.paddingRight - set the padding between the parent's right edge and children.
- *treemap*.paddingBottom - set the padding between the parent's bottom edge and children.
- *treemap*.paddingLeft - set the padding between the parent's left edge and children.
- d3.treemapBinary - tile using a balanced binary tree.
- d3.treemapDice - tile into a horizontal row.
- d3.treemapSlice - tile into a vertical column.
- d3.treemapSliceDice - alternate between slicing and dicing.
- d3.treemapSquarify - tile using squarified rows per Bruls *et. al.*
- d3.treemapResquarify - like d3.treemapSquarify, but performs stable updates.
- *squarify*.ratio - set the desired rectangle aspect ratio.
- d3.partition - create a new partition (icicle or sunburst) layout.
- *partition* - layout the specified hierarchy as a partition diagram.
- *partition*.size - set the layout size.
- *partition*.round - set whether the output coordinates are rounded.
- *partition*.padding - set the padding.
- d3.pack - create a new circle-packing layout.
- *pack* - layout the specified hierarchy using circle-packing.

- *pack*.radius - set the radius accessor.
- *pack*.size - set the layout size.
- *pack*.padding - set the padding.
- d3.packSiblings - pack the specified array of circles.
- d3.packEnclose - enclose the specified array of circles.

Interpolators (d3-interpolate)

Interpolate numbers, colors, strings, arrays, objects, whatever!

- d3.interpolate - interpolate arbitrary values.
- d3.interpolateNumber - interpolate numbers.
- d3.interpolateRound - interpolate integers.
- d3.interpolateString - interpolate strings with embedded numbers.
- d3.interpolateDate - interpolate dates.
- d3.interpolateArray - interpolate arrays of arbitrary values.
- d3.interpolateNumberArray - interpolate arrays of numbers.
- d3.interpolateObject - interpolate arbitrary objects.
- d3.interpolateTransformCss - interpolate 2D CSS transforms.
- d3.interpolateTransformSvg - interpolate 2D SVG transforms.
- d3.interpolateZoom - zoom and pan between two views.
- *interpolateZoom*.rho - set the curvature *rho* of the zoom interpolator.
- d3.interpolateDiscrete - generate a discrete interpolator from a set of values.
- d3.quantize - generate uniformly-spaced samples from an interpolator.
- d3.interpolateRgb - interpolate RGB colors.
- d3.interpolateRgbBasis - generate a B-spline through a set of colors.
- d3.interpolateRgbBasisClosed - generate a closed B-spline through a set of colors.
- d3.interpolateHsl - interpolate HSL colors.
- d3.interpolateHslLong - interpolate HSL colors, the long way.
- d3.interpolateLab - interpolate Lab colors.
- d3.interpolateHcl - interpolate HCL colors.
- d3.interpolateHclLong - interpolate HCL colors, the long way.
- d3.interpolateCubehelix - interpolate Cubehelix colors.
- d3.interpolateCubehelixLong - interpolate Cubehelix colors, the long way.
- *interpolate*.gamma - apply gamma correction during interpolation.
- d3.interpolateHue - interpolate a hue angle.
- d3.interpolateBasis - generate a B-spline through a set of values.
- d3.interpolateBasisClosed - generate a closed B-spline through a set of values.
- d3.piecewise - generate a piecewise linear interpolator from a set of values.

Paths (d3-path)

Serialize Canvas path commands to SVG.

- d3.path - create a new path serializer.

- *path*.moveTo - move to the given point.
- *path*.closePath - close the current subpath.
- *path*.lineTo - draw a straight line segment.
- *path*.quadraticCurveTo - draw a quadratic Bézier segment.
- *path*.bezierCurveTo - draw a cubic Bézier segment.
- *path*.arcTo - draw a circular arc segment.
- *path*.arc - draw a circular arc segment.
- *path*.rect - draw a rectangle.
- *path*.toString - serialize to an SVG path data string.

Polygons (d3-polygon)

Geometric operations for two-dimensional polygons.

- d3.polygonArea - compute the area of the given polygon.
- d3.polygonCentroid - compute the centroid of the given polygon.
- d3.polygonHull - compute the convex hull of the given points.
- d3.polygonContains - test whether a point is inside a polygon.
- d3.polygonLength - compute the length of the given polygon's perimeter.

Quadtrees (d3-quadtree)

Two-dimensional recursive spatial subdivision.

- d3.quadtree - create a new, empty quadtree.
- *quadtree*.x - set the x accessor.
- *quadtree*.y - set the y accessor.
- *quadtree*.extent - extend the quadtree to cover an extent.
- *quadtree*.cover - extend the quadtree to cover a point.
- *quadtree*.add - add a datum to a quadtree.
- *quadtree*.addAll - add an array of data to a quadtree.
- *quadtree*.remove - remove a datum from a quadtree.
- *quadtree*.removeAll - remove an array of data from a quadtree.
- *quadtree*.copy - create a copy of a quadtree.
- *quadtree*.root - get the quadtree's root node.
- *quadtree*.data - retrieve all data from the quadtree.
- *quadtree*.size - count the number of data in the quadtree.
- *quadtree*.find - quickly find the closest datum in a quadtree.
- *quadtree*.visit - selectively visit nodes in a quadtree.
- *quadtree*.visitAfter - visit all nodes in a quadtree.

Random Numbers (d3-random)

Generate random numbers from various distributions.

- d3.randomUniform - from a uniform distribution.
- d3.randomInt - from a uniform integer distribution.

- d3.randomNormal - from a normal distribution.
- d3.randomLogNormal - from a log-normal distribution.
- d3.randomBates - from a Bates distribution.
- d3.randomIrwinHall - from an Irwin–Hall distribution.
- d3.randomExponential - from an exponential distribution.
- d3.randomPareto - from a Pareto distribution.
- d3.randomBernoulli - from a Bernoulli distribution.
- d3.randomGeometric - from a geometric distribution.
- d3.randomBinomial - from a binomial distribution.
- d3.randomGamma - from a gamma distribution.
- d3.randomBeta - from a beta distribution.
- d3.randomWeibull - from a Weibull, Gumbel or Fréchet distribution.
- d3.randomCauchy - from a Cauchy distribution.
- d3.randomLogistic - from a logistic distribution.
- d3.randomPoisson - from a Poisson distribution.
- *random*.source - set the source of randomness.
- d3.randomLcg - a seeded pseudorandom number generator.

Scales (d3-scale)

Encodings that map abstract data to visual representation.

Continuous Scales

Map a continuous, quantitative domain to a continuous range.

- *continuous* - compute the range value corresponding to a given domain value.
- *continuous*.invert - compute the domain value corresponding to a given range value.
- *continuous*.domain - set the input domain.
- *continuous*.range - set the output range.
- *continuous*.rangeRound - set the output range and enable rounding.
- *continuous*.clamp - enable clamping to the domain or range.
- *continuous*.unknown - set the output value for unknown inputs.
- *continuous*.interpolate - set the output interpolator.
- *continuous*.ticks - compute representative values from the domain.
- *continuous*.tickFormat - format ticks for human consumption.
- *continuous*.nice - extend the domain to nice round numbers.
- *continuous*.copy - create a copy of this scale.
- d3.tickFormat - format ticks for human consumption.
- d3.scaleLinear - create a quantitative linear scale.
- d3.scalePow - create a quantitative power scale.
- *pow* - compute the range value corresponding to a given domain value.
- *pow*.invert - compute the domain value corresponding to a given range value.
- *pow*.exponent - set the power exponent.

- *pow*.domain - set the input domain.
- *pow*.range - set the output range.
- *pow*.rangeRound - set the output range and enable rounding.
- *pow*.clamp - enable clamping to the domain or range.
- *pow*.interpolate - set the output interpolator.
- *pow*.ticks - compute representative values from the domain.
- *pow*.tickFormat - format ticks for human consumption.
- *pow*.nice - extend the domain to nice round numbers.
- *pow*.copy - create a copy of this scale.
- d3.scaleSqrt - create a quantitative power scale with exponent 0.5.
- d3.scaleLog - create a quantitative logarithmic scale.
- *log* - compute the range value corresponding to a given domain value.
- *log*.invert - compute the domain value corresponding to a given range value.
- *log*.base - set the logarithm base.
- *log*.domain - set the input domain.
- *log*.range - set the output range.
- *log*.rangeRound - set the output range and enable rounding.
- *log*.clamp - enable clamping to the domain or range.
- *log*.interpolate - set the output interpolator.
- *log*.ticks - compute representative values from the domain.
- *log*.tickFormat - format ticks for human consumption.
- *log*.nice - extend the domain to nice round numbers.
- *log*.copy - create a copy of this scale.
- d3.scaleSymlog - create a symmetric logarithmic scale.
- *symlog*.constant - set the constant of a symlog scale.
- d3.scaleIdentity - creates an identity scale.
- d3.scaleRadial - creates a radial scale.
- d3.scaleTime - create a linear scale for time.
- *time* - compute the range value corresponding to a given domain value.
- *time*.invert - compute the domain value corresponding to a given range value.
- *time*.domain - set the input domain.
- *time*.range - set the output range.
- *time*.rangeRound - set the output range and enable rounding.
- *time*.clamp - enable clamping to the domain or range.
- *time*.interpolate - set the output interpolator.
- *time*.ticks - compute representative values from the domain.
- *time*.tickFormat - format ticks for human consumption.
- *time*.nice - extend the domain to nice round times.
- *time*.copy - create a copy of this scale.
- d3.scaleUtc - create a linear scale for UTC.

Sequential Scales

Map a continuous, quantitative domain to a continuous, fixed interpolator.

- d3.scaleSequential - create a sequential scale.
- *sequential* - compute the range value corresponding to an input value.
- *sequential*.domain - set the input domain.
- *sequential*.clamp - enable clamping to the domain.
- *sequential*.interpolator - set the scale's output interpolator.
- *sequential*.range - set the output range.
- *sequential*.rangeRound - set the output range and enable rounding.
- *sequential*.copy - create a copy of this scale.
- d3.scaleSequentialLog - create a logarithmic sequential scale.
- d3.scaleSequentialPow - create a power sequential scale.
- d3.scaleSequentialSqrt - create a power sequential scale with exponent 0.5.
- d3.scaleSequentialSymlog - create a symmetric logarithmic sequential scale.
- d3.scaleSequentialQuantile - create a sequential scale using a p-quantile transform.
- *sequentialQuantile*.quantiles - return the scale's quantiles.

Diverging Scales

Map a continuous, quantitative domain to a continuous, fixed interpolator.

- d3.scaleDiverging - create a diverging scale.
- *diverging* - compute the range value corresponding to an input value.
- *diverging*.domain - set the input domain.
- *diverging*.clamp - enable clamping to the domain or range.
- *diverging*.interpolator - set the scale's output interpolator.
- *diverging*.range - set the output range.
- *diverging*.rangeRound - set the output range and enable rounding.
- *diverging*.copy - create a copy of this scale.
- *diverging*.unknown - set the output value for unknown inputs.
- d3.scaleDivergingLog - create a diverging logarithmic scale.
- d3.scaleDivergingPow - create a diverging power scale.
- d3.scaleDivergingSqrt - create a diverging power scale with exponent 0.5.
- d3.scaleDivergingSymlog - create a diverging symmetric logarithmic scale.

Quantize Scales

Map a continuous, quantitative domain to a discrete range.

- d3.scaleQuantize - create a uniform quantizing linear scale.
- *quantize* - compute the range value corresponding to a given domain value.
- *quantize*.invertExtent - compute the domain values corresponding to a given range value.
- *quantize*.domain - set the input domain.
- *quantize*.range - set the output range.
- *quantize*.ticks - compute representative values from the domain.
- *quantize*.tickFormat - format ticks for human consumption.
- *quantize*.nice - extend the domain to nice round numbers.

- *quantize*.thresholds - return the array of computed thresholds within the domain.
- *quantize*.copy - create a copy of this scale.
- d3.scaleQuantile - create a quantile quantizing linear scale.
- *quantile* - compute the range value corresponding to a given domain value.
- *quantile*.invertExtent - compute the domain values corresponding to a given range value.
- *quantile*.domain - set the input domain.
- *quantile*.range - set the output range.
- *quantile*.quantiles - get the quantile thresholds.
- *quantile*.copy - create a copy of this scale.
- d3.scaleThreshold - create an arbitrary quantizing linear scale.
- *threshold* - compute the range value corresponding to a given domain value.
- *threshold*.invertExtent - compute the domain values corresponding to a given range value.
- *threshold*.domain - set the input domain.
- *threshold*.range - set the output range.
- *threshold*.copy - create a copy of this scale.

Ordinal Scales

Map a discrete domain to a discrete range.

- d3.scaleOrdinal - create an ordinal scale.
- *ordinal* - compute the range value corresponding to a given domain value.
- *ordinal*.domain - set the input domain.
- *ordinal*.range - set the output range.
- *ordinal*.unknown - set the output value for unknown inputs.
- *ordinal*.copy - create a copy of this scale.
- d3.scaleImplicit - a special unknown value for implicit domains.
- d3.scaleBand - create an ordinal band scale.
- *band* - compute the band start corresponding to a given domain value.
- *band*.domain - set the input domain.
- *band*.range - set the output range.
- *band*.rangeRound - set the output range and enable rounding.
- *band*.round - enable rounding.
- *band*.paddingInner - set padding between bands.
- *band*.paddingOuter - set padding outside the first and last bands.
- *band*.padding - set padding outside and between bands.
- *band*.align - set band alignment, if there is extra space.
- *band*.bandwidth - get the width of each band.
- *band*.step - get the distance between the starts of adjacent bands.
- *band*.copy - create a copy of this scale.
- d3.scalePoint - create an ordinal point scale.
- *point* - compute the point corresponding to a given domain value.
- *point*.domain - set the input domain.

- *point*.range - set the output range.
- *point*.rangeRound - set the output range and enable rounding.
- *point*.round - enable rounding.
- *point*.padding - set padding outside the first and last point.
- *point*.align - set point alignment, if there is extra space.
- *point*.bandwidth - returns zero.
- *point*.step - get the distance between the starts of adjacent points.
- *point*.copy - create a copy of this scale.

Selections (d3-selection)

Transform the DOM by selecting elements and joining to data.

Selecting Elements

- d3.selection - select the root document element.
- d3.select - select an element from the document.
- d3.selectAll - select multiple elements from the document.
- *selection*.select - select a descendant element for each selected element.
- *selection*.selectAll - select multiple descendants for each selected element.
- *selection*.filter - filter elements based on data.
- *selection*.merge - merge this selection with another.
- *selection*.selectChild - select a child element for each selected element.
- *selection*.selectChildren - select the children elements for each selected element.
- *selection*.selection - return the selection.
- d3.matcher - test whether an element matches a selector.
- d3.selector - select an element.
- d3.selectorAll - select elements.
- d3.window - get a node's owner window.
- d3.style - get a node's current style value.

Modifying Elements

- *selection*.attr - get or set an attribute.
- *selection*.classed - get, add or remove CSS classes.
- *selection*.style - get or set a style property.
- *selection*.property - get or set a (raw) property.
- *selection*.text - get or set the text content.
- *selection*.html - get or set the inner HTML.
- *selection*.append - create, append and select new elements.
- *selection*.insert - create, insert and select new elements.
- *selection*.remove - remove elements from the document.
- *selection*.clone - insert clones of selected elements.
- *selection*.sort - sort elements in the document based on data.
- *selection*.order - reorders elements in the document to match the selection.

- *selection*.raise - reorders each element as the last child of its parent.
- *selection*.lower - reorders each element as the first child of its parent.
- d3.create - create and select a detached element.
- d3.creator - create an element by name.

Joining Data

- *selection*.data - bind elements to data.
- *selection*.join - enter, update or exit elements based on data.
- *selection*.enter - get the enter selection (data missing elements).
- *selection*.exit - get the exit selection (elements missing data).
- *selection*.datum - get or set element data (without joining).

Handling Events

- *selection*.on - add or remove event listeners.
- *selection*.dispatch - dispatch a custom event.
- d3.pointer - get the pointer's position of an event.
- d3.pointers - get the pointers' positions of an event.

Control Flow

- *selection*.each - call a function for each element.
- *selection*.call - call a function with this selection.
- *selection*.empty - returns true if this selection is empty.
- *selection*.nodes - returns an array of all selected elements.
- *selection*.node - returns the first (non-null) element.
- *selection*.size - returns the count of elements.
- *selection*[Symbol.iterator] - iterate over the selection's nodes.

Local Variables

- d3.local - declares a new local variable.
- *local*.set - set a local variable's value.
- *local*.get - get a local variable's value.
- *local*.remove - delete a local variable.
- *local*.toString - get the property identifier of a local variable.

Namespaces

- d3.namespace - qualify a prefixed XML name, such as "xlink:href".
- d3.namespaces - the built-in XML namespaces.

Shapes (d3-shape)

Graphical primitives for visualization.

Arcs

Circular or annular sectors, as in a pie or donut chart.

- d3.arc - create a new arc generator.
- *arc* - generate an arc for the given datum.
- *arc*.centroid - compute an arc's midpoint.
- *arc*.innerRadius - set the inner radius.
- *arc*.outerRadius - set the outer radius.
- *arc*.cornerRadius - set the corner radius, for rounded corners.
- *arc*.startAngle - set the start angle.
- *arc*.endAngle - set the end angle.
- *arc*.padAngle - set the angle between adjacent arcs, for padded arcs.
- *arc*.padRadius - set the radius at which to linearize padding.
- *arc*.context - set the rendering context.

Pies

Compute the necessary angles to represent a tabular dataset as a pie or donut chart.

- d3.pie - create a new pie generator.
- *pie* - compute the arc angles for the given dataset.
- *pie*.value - set the value accessor.
- *pie*.sort - set the sort order comparator.
- *pie*.sortValues - set the sort order comparator.
- *pie*.startAngle - set the overall start angle.
- *pie*.endAngle - set the overall end angle.
- *pie*.padAngle - set the pad angle between adjacent arcs.

Lines

A spline or polyline, as in a line chart.

- d3.line - create a new line generator.
- *line* - generate a line for the given dataset.
- *line*.x - set the x accessor.
- *line*.y - set the y accessor.
- *line*.defined - set the defined accessor.
- *line*.curve - set the curve interpolator.
- *line*.context - set the rendering context.
- d3.lineRadial - create a new radial line generator.
- *lineRadial* - generate a line for the given dataset.
- *lineRadial*.angle - set the angle accessor.
- *lineRadial*.radius - set the radius accessor.
- *lineRadial*.defined - set the defined accessor.
- *lineRadial*.curve - set the curve interpolator.
- *lineRadial*.context - set the rendering context.

Areas

An area, defined by a bounding topline and baseline, as in an area chart.

- d3.area - create a new area generator.
- *area* - generate an area for the given dataset.
- *area*.x - set the *x0* and *x1* accessors.
- *area*.x0 - set the baseline x accessor.
- *area*.x1 - set the topline x accessor.
- *area*.y - set the *y0* and *y1* accessors.
- *area*.y0 - set the baseline y accessor.
- *area*.y1 - set the topline y accessor.
- *area*.defined - set the defined accessor.
- *area*.curve - set the curve interpolator.
- *area*.context - set the rendering context.
- *area*.lineX0 - derive a line for the left edge of an area.
- *area*.lineY0 - derive a line for the top edge of an area.
- *area*.lineX1 - derive a line for the right edge of an area.
- *area*.lineY1 - derive a line for the bottom edge of an area.
- d3.areaRadial - create a new radial area generator.
- *areaRadial* - generate an area for the given dataset.
- *areaRadial*.angle - set the start and end angle accessors.
- *areaRadial*.startAngle - set the start angle accessor.
- *areaRadial*.endAngle - set the end angle accessor.
- *areaRadial*.radius - set the inner and outer radius accessors.
- *areaRadial*.innerRadius - set the inner radius accessor.
- *areaRadial*.outerRadius - set the outer radius accessor.
- *areaRadial*.defined - set the defined accessor.
- *areaRadial*.curve - set the curve interpolator.
- *areaRadial*.context - set the rendering context.
- *areaRadial*.lineStartAngle - derive a line for the start edge of an area.
- *areaRadial*.lineInnerRadius - derive a line for the inner edge of an area.
- *areaRadial*.lineEndAngle - derive a line for the end edge of an area.
- *areaRadial*.lineOuterRadius - derive a line for the outer edge of an area.

Curves

Interpolate between points to produce a continuous shape.

- d3.curveBasis - a cubic basis spline, repeating the end points.
- d3.curveBasisClosed - a closed cubic basis spline.
- d3.curveBasisOpen - a cubic basis spline.
- d3.curveBundle - a straightened cubic basis spline.
- *bundle*.beta - set the bundle tension *beta*.
- d3.curveBumpX - a cubic Bézier spline with horizontal tangents.
- d3.curveBumpY - a cubic Bézier spline with vertical tangents.

- d3.curveCardinal - a cubic cardinal spline, with one-sided difference at each end.
- d3.curveCardinalClosed - a closed cubic cardinal spline.
- d3.curveCardinalOpen - a cubic cardinal spline.
- *cardinal*.tension - set the cardinal spline tension.
- d3.curveCatmullRom - a cubic Catmull–Rom spline, with one-sided difference at each end.
- d3.curveCatmullRomClosed - a closed cubic Catmull–Rom spline.
- d3.curveCatmullRomOpen - a cubic Catmull–Rom spline.
- *catmullRom*.alpha - set the Catmull–Rom parameter *alpha*.
- d3.curveLinear - a polyline.
- d3.curveLinearClosed - a closed polyline.
- d3.curveMonotoneX - a cubic spline that, given monotonicity in x, preserves it in y.
- d3.curveMonotoneY - a cubic spline that, given monotonicity in y, preserves it in x.
- d3.curveNatural - a natural cubic spline.
- d3.curveStep - a piecewise constant function.
- d3.curveStepAfter - a piecewise constant function.
- d3.curveStepBefore - a piecewise constant function.
- *curve*.areaStart - start a new area segment.
- *curve*.areaEnd - end the current area segment.
- *curve*.lineStart - start a new line segment.
- *curve*.lineEnd - end the current line segment.
- *curve*.point - add a point to the current line segment.

Links

A smooth cubic Bézier curve from a source to a target.

- d3.link - create a new link generator.
- d3.linkVertical - create a new vertical link generator.
- d3.linkHorizontal - create a new horizontal link generator.
- *link* - generate a link.
- *link*.source - set the source accessor.
- *link*.target - set the target accessor.
- *link*.x - set the point x-accessor.
- *link*.y - set the point y-accessor.
- *link*.context - set the rendering context.
- d3.linkRadial - create a new radial link generator.
- *linkRadial*.angle - set the point *angle* accessor.
- *linkRadial*.radius - set the point *radius* accessor.

Symbols

A categorical shape encoding, as in a scatterplot.

- d3.symbol - create a new symbol generator.
- *symbol* - generate a symbol for the given datum.
- *symbol*.type - set the symbol type.
- *symbol*.size - set the size of the symbol in square pixels.
- *symbol*.context - set the rendering context.
- d3.symbolsFill - an array of built-in symbol types for filling.
- d3.symbolsStroke - an array of built-in symbol types for stroking.
- d3.symbolAsterisk - an asterisk; for stroke.
- d3.symbolCircle - a circle; for fill or stroke.
- d3.symbolCross - a Greek cross with arms of equal length; for fill.
- d3.symbolDiamond - a rhombus; for fill.
- d3.symbolDiamond2 - a rotated square; for stroke.
- d3.symbolPlus - a plus sign; for stroke.
- d3.symbolSquare - a square; for fill.
- d3.symbolSquare2 - a square; for stroke.
- d3.symbolStar - a pentagonal star (pentagram); for fill.
- d3.symbolTriangle - an up-pointing triangle; for fill.
- d3.symbolTriangle2 - an up-pointing triangle; for stroke.
- d3.symbolWye - a Y shape; for fill.
- d3.symbolX - an X shape; for stroke.
- d3.pointRadial - relative coordinates of a point given an angle and radius.
- *symbolType*.draw - draw this symbol to the given context.

Stacks

Stack shapes, placing one adjacent to another, as in a stacked bar chart.

- d3.stack - create a new stack generator.
- *stack* - generate a stack for the given dataset.
- *stack*.keys - set the keys accessor.
- *stack*.value - set the value accessor.
- *stack*.order - set the order accessor.
- *stack*.offset - set the offset accessor.
- d3.stackOrderAppearance - put the earliest series on bottom.
- d3.stackOrderAscending - put the smallest series on bottom.
- d3.stackOrderDescending - put the largest series on bottom.
- d3.stackOrderInsideOut - put earlier series in the middle.
- d3.stackOrderNone - use the given series order.
- d3.stackOrderReverse - use the reverse of the given series order.
- d3.stackOffsetExpand - normalize the baseline to zero and topline to one.
- d3.stackOffsetDiverging - positive above zero; negative below zero.
- d3.stackOffsetNone - apply a zero baseline.
- d3.stackOffsetSilhouette - center the streamgraph around zero.
- d3.stackOffsetWiggle - minimize streamgraph wiggling.

Time Formats (d3-time-format)

Parse and format times, inspired by strptime and strftime.

- d3.timeFormat - alias for *locale*.format on the default locale.
- d3.timeParse - alias for *locale*.parse on the default locale.
- d3.utcFormat - alias for *locale*.utcFormat on the default locale.
- d3.utcParse - alias for *locale*.utcParse on the default locale.
- d3.isoFormat - an ISO 8601 UTC formatter.
- d3.isoParse - an ISO 8601 UTC parser.
- *locale*.format - create a time formatter.
- *locale*.parse - create a time parser.
- *locale*.utcFormat - create a UTC formatter.
- *locale*.utcParse - create a UTC parser.
- d3.timeFormatLocale - define a custom locale.
- d3.timeFormatDefaultLocale - define the default locale.

Time Intervals (d3-time)

A calculator for humanity's peculiar conventions of time.

- d3.timeInterval - implement a new custom time interval.
- *interval* - alias for *interval*.floor.
- *interval*.floor - round down to the nearest boundary.
- *interval*.round - round to the nearest boundary.
- *interval*.ceil - round up to the nearest boundary.
- *interval*.offset - offset a date by some number of intervals.
- *interval*.range - generate a range of dates at interval boundaries.
- *interval*.filter - create a filtered subset of this interval.
- *interval*.every - create a filtered subset of this interval.
- *interval*.count - count interval boundaries between two dates.
- d3.timeMillisecond, d3.utcMillisecond - the millisecond interval.
- d3.timeMilliseconds, d3.utcMilliseconds - aliases for millisecond.range.
- d3.timeSecond, d3.utcSecond - the second interval.
- d3.timeSeconds, d3.utcSeconds - aliases for second.range.
- d3.timeMinute, d3.utcMinute - the minute interval.
- d3.timeMinutes, d3.utcMinutes - aliases for minute.range.
- d3.timeHour, d3.utcHour - the hour interval.
- d3.timeHours, d3.utcHours - aliases for hour.range.
- d3.timeDay, d3.utcDay - the day interval.
- d3.timeDays, d3.utcDays - aliases for day.range.
- d3.timeWeek, d3.utcWeek - aliases for sunday.
- d3.timeWeeks, d3.utcWeeks - aliases for week.range.
- d3.timeSunday, d3.utcSunday - the week interval, starting on Sunday.
- d3.timeSundays, d3.utcSundays - aliases for sunday.range.
- d3.timeMonday, d3.utcMonday - the week interval, starting on Monday.
- d3.timeMondays, d3.utcMondays - aliases for monday.range.

- d3.timeTuesday, d3.utcTuesday - the week interval, starting on Tuesday.
- d3.timeTuesdays, d3.utcTuesdays - aliases for tuesday.range.
- d3.timeWednesday, d3.utcWednesday - the week interval, starting on Wednesday.
- d3.timeWednesdays, d3.utcWednesdays - aliases for wednesday.range.
- d3.timeThursday, d3.utcThursday - the week interval, starting on Thursday.
- d3.timeThursdays, d3.utcThursdays - aliases for thursday.range.
- d3.timeFriday, d3.utcFriday - the week interval, starting on Friday.
- d3.timeFridays, d3.utcFridays - aliases for friday.range.
- d3.timeSaturday, d3.utcSaturday - the week interval, starting on Saturday.
- d3.timeSaturdays, d3.utcSaturdays - aliases for saturday.range.
- d3.timeMonth, d3.utcMonth - the month interval.
- d3.timeMonths, d3.utcMonths - aliases for month.range.
- d3.timeYear, d3.utcYear - the year interval.
- d3.timeYears, d3.utcYears - aliases for year.range.
- d3.timeTicks, d3.utcTicks -
- d3.timeTickInterval, d3.utcTickInterval -

Timers (d3-timer)

An efficient queue for managing thousands of concurrent animations.

- d3.now - get the current high-resolution time.
- d3.timer - schedule a new timer.
- *timer*.restart - reset the timer's start time and callback.
- *timer*.stop - stop the timer.
- d3.timerFlush - immediately execute any eligible timers.
- d3.timeout - schedule a timer that stops on its first callback.
- d3.interval - schedule a timer that is called with a configurable period.

Transitions (d3-transition)

Animated transitions for selections.

- *selection*.transition - schedule a transition for the selected elements.
- *selection*.interrupt - interrupt and cancel transitions on the selected elements.
- d3.interrupt - interrupt the active transition for a given node.
- d3.transition - schedule a transition on the root document element.
- *transition*.select - schedule a transition on the selected elements.
- *transition*.selectAll - schedule a transition on the selected elements.
- *transition*.selectChild - select a child element for each selected element.
- *transition*.selectChildren - select the children elements for each selected element.
- *transition*.selection - returns a selection for this transition.
- *transition*.filter - filter elements based on data.
- *transition*.merge - merge this transition with another.

- *transition*.transition - schedule a new transition following this one.
- d3.active - select the active transition for a given node.
- *transition*.attr - tween the given attribute using the default interpolator.
- *transition*.attrTween - tween the given attribute using a custom interpolator.
- *transition*.style - tween the given style property using the default interpolator.
- *transition*.styleTween - tween the given style property using a custom interpolator.
- *transition*.text - set the text content when the transition starts.
- *transition*.textTween - tween the text using a custom interpolator.
- *transition*.remove - remove the selected elements when the transition ends.
- *transition*.tween - run custom code during the transition.
- *transition*.delay - specify per-element delay in milliseconds.
- *transition*.duration - specify per-element duration in milliseconds.
- *transition*.ease - specify the easing function.
- *transition*.easeVarying - specify an easing function factory.
- *transition*.end - a promise that resolves when a transition ends.
- *transition*.on - await the end of a transition.
- *transition*.each - call a function for each element.
- *transition*.call - call a function with this transition.
- *transition*.empty - returns true if this transition is empty.
- *transition*.nodes - returns an array of all selected elements.
- *transition*.node - returns the first (non-null) element.
- *transition*.size - returns the count of elements.

Zooming (d3-zoom)

Pan and zoom SVG, HTML or Canvas using mouse or touch input.

- d3.zoom - create a zoom behavior.
- *zoom* - apply the zoom behavior to the selected elements.
- *zoom*.transform - change the transform for the selected elements.
- *zoom*.translateBy - translate the transform for the selected elements.
- *zoom*.translateTo - translate the transform for the selected elements.
- *zoom*.scaleBy - scale the transform for the selected elements.
- *zoom*.scaleTo - scale the transform for the selected elements.
- *zoom*.constrain - override the transform constraint logic.
- *zoom*.filter - control which input events initiate zooming.
- *zoom*.touchable - set the touch support detector.
- *zoom*.wheelDelta - override scaling for wheel events.
- *zoom*.extent - set the extent of the viewport.
- *zoom*.scaleExtent - set the allowed scale range.
- *zoom*.translateExtent - set the extent of the zoomable world.
- *zoom*.clickDistance - set the click distance threshold.
- *zoom*.tapDistance - set the tap distance threshold.

- *zoom*.duration - set the duration of zoom transitions.
- *zoom*.interpolate - control the interpolation of zoom transitions.
- *zoom*.on - listen for zoom events.
- d3.zoomTransform - get the zoom transform for a given element.
- *transform*.scale - scale a transform by the specified amount.
- *transform*.translate - translate a transform by the specified amount.
- *transform*.apply - apply the transform to the given point.
- *transform*.applyX - apply the transform to the given x-coordinate.
- *transform*.applyY - apply the transform to the given y-coordinate.
- *transform*.invert - unapply the transform to the given point.
- *transform*.invertX - unapply the transform to the given x-coordinate.
- *transform*.invertY - unapply the transform to the given y-coordinate.
- *transform*.rescaleX - apply the transform to an x-scale's domain.
- *transform*.rescaleY - apply the transform to a y-scale's domain.
- *transform*.toString - format the transform as an SVG transform string.
- d3.zoomIdentity - the identity transform.