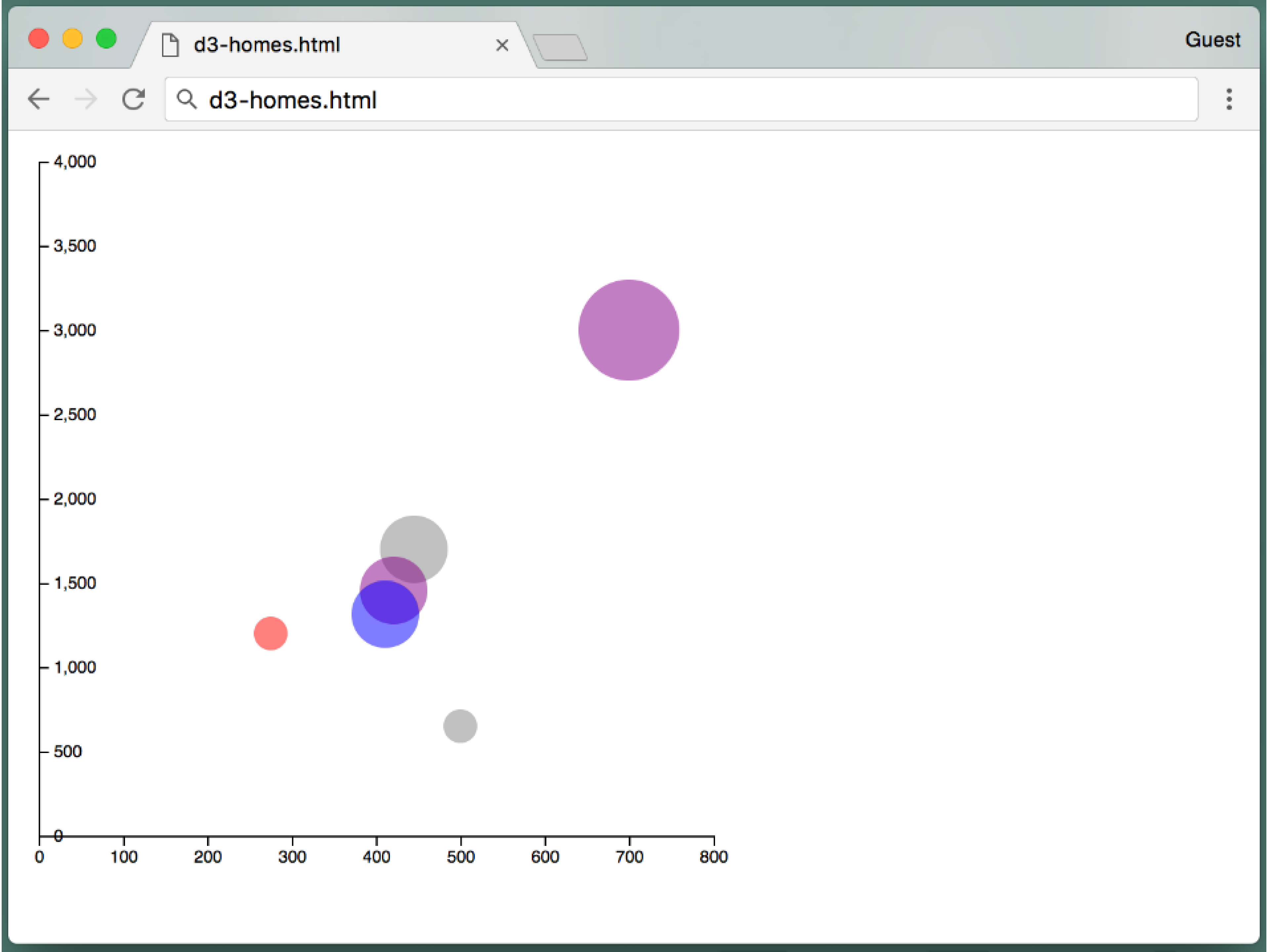
Video 3.11

More D3

Chris Murphy

Review

- D3.js allows us to generate HTML and SVG elements based on data
- We can apply functions to data sets to generate graphical elements, e.g. charts


d3-homes.html
Guest
d3-homes.html
4,000
3,500
3,000
2,500
2,000
1,500
1,000
500
0
0
100
200
300
400
500
600
700
800

D3 and Data

- The data used in D3 can be objects, not just numbers
- We can then use the properties of these objects when deciding how to render the visualization (chart, SVG, etc.)

```
<html>
<head>
<script src="http://d3js.org/d3.v4.min.js"></script>
</head>

<body>
<svg class="chart" height="900" width="900">
</svg>

<script>

var values = [
 {price: 700, sqft: 3000, br: 3, pets: [ 'cats', 'dogs' ] },
 {price: 445, sqft: 1700, br: 2, pets: [] },
 {price: 421, sqft: 1455, br: 2, pets: [ 'cats', 'dogs' ] },
 {price: 411, sqft: 1314, br: 2, pets: [ 'dogs' ] },
 {price: 275, sqft: 1200, br: 1, pets: [ 'cats' ]},
 {price: 500, sqft: 650, br: 1, pets: [] },
];

. . .
```

```
<html>
<head>
<script src="http://d3js.org/d3.v4.min.js"></script>
</head>

<body>
<svg class="chart" height="900" width="900">
</svg>

<script>

var values = [
 {price: 700, sqft: 3000, br: 3, pets: [ 'cats', 'dogs' ] },
 {price: 445, sqft: 1700, br: 2, pets: [] },
 {price: 421, sqft: 1455, br: 2, pets: [ 'cats', 'dogs' ] },
 {price: 411, sqft: 1314, br: 2, pets: [ 'dogs' ] },
 {price: 275, sqft: 1200, br: 1, pets: [ 'cats' ]},
 {price: 500, sqft: 650, br: 1, pets: [] },
];

. . .
```

```
<html>
<head>
<script src="http://d3js.org/d3.v4.min.js"></script>
</head>

<body>
<svg class="chart" height="900" width="900">
</svg>

<script>

var values = [
 {price: 700, sqft: 3000, br: 3, pets: [ 'cats', 'dogs' ] },
 {price: 445, sqft: 1700, br: 2, pets: [] },
 {price: 421, sqft: 1455, br: 2, pets: [ 'cats', 'dogs' ] },
 {price: 411, sqft: 1314, br: 2, pets: [ 'dogs' ] },
 {price: 275, sqft: 1200, br: 1, pets: [ 'cats' ]},
 {price: 500, sqft: 650, br: 1, pets: [] },
];

. . .
```

```
<html>
<head>
<script src="http://d3js.org/d3.v4.min.js"></script>
</head>

<body>
<svg class="chart" height="900" width="900">
</svg>

<script>

var values = [
 {price: 700, sqft: 3000, br: 3, pets: [ 'cats', 'dogs' ] },
 {price: 445, sqft: 1700, br: 2, pets: [] },
 {price: 421, sqft: 1455, br: 2, pets: [ 'cats', 'dogs' ] },
 {price: 411, sqft: 1314, br: 2, pets: [ 'dogs' ] },
 {price: 275, sqft: 1200, br: 1, pets: [ 'cats' ]},
 {price: 500, sqft: 650, br: 1, pets: [] },
];

. . .
```

```
<html>
<head>
<script src="http://d3js.org/d3.v4.min.js"></script>
</head>

<body>
<svg class="chart" height="900" width="900">
</svg>

<script>

var values = [
 {price: 700, sqft: 3000, br: 3, pets: [ 'cats', 'dogs' ] },
 {price: 445, sqft: 1700, br: 2, pets: [] },
 {price: 421, sqft: 1455, br: 2, pets: [ 'cats', 'dogs' ] },
 {price: 411, sqft: 1314, br: 2, pets: [ 'dogs' ] },
 {price: 275, sqft: 1200, br: 1, pets: [ 'cats' ]},
 {price: 500, sqft: 650, br: 1, pets: [] },
];

. . .
```

```
<html>
<head>
<script src="http://d3js.org/d3.v4.min.js"></script>
</head>

<body>
<svg class="chart" height="900" width="900">
</svg>

<script>

var values = [
 {price: 700, sqft: 3000, br: 3, pets: [ 'cats', 'dogs' ] },
 {price: 445, sqft: 1700, br: 2, pets: [] },
 {price: 421, sqft: 1455, br: 2, pets: [ 'cats', 'dogs' ] },
 {price: 411, sqft: 1314, br: 2, pets: [ 'dogs' ] },
 {price: 275, sqft: 1200, br: 1, pets: [ 'cats' ]},
 {price: 500, sqft: 650, br: 1, pets: [] },
];

. . .
```

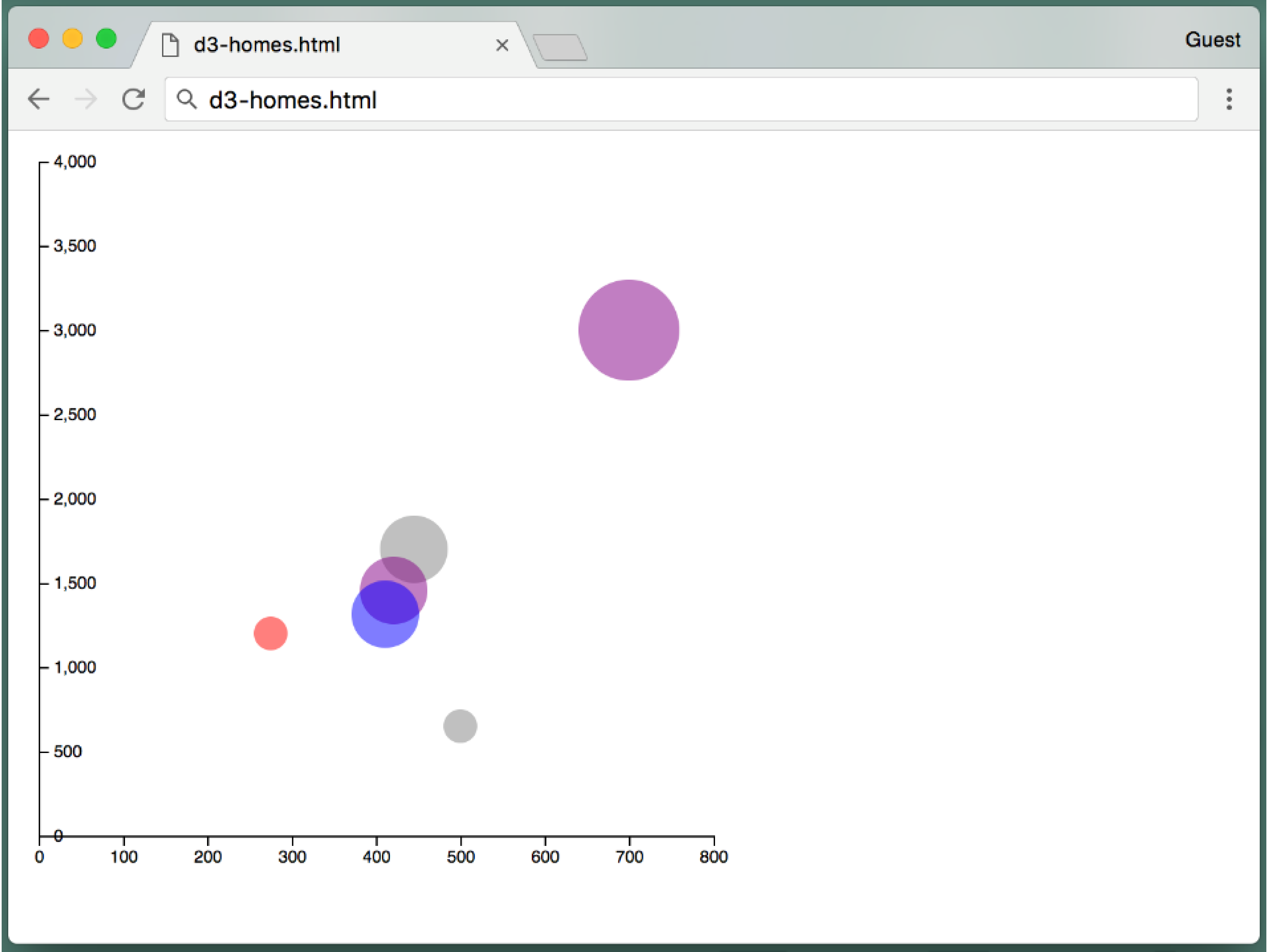
```
<html>
<head>
<script src="http://d3js.org/d3.v4.min.js"></script>
</head>

<body>
<svg class="chart" height="900" width="900">
</svg>

<script>

var values = [
 {price: 700, sqft: 3000, br: 3, pets: [ 'cats', 'dogs' ] },
 {price: 445, sqft: 1700, br: 2, pets: [] },
 {price: 421, sqft: 1455, br: 2, pets: [ 'cats', 'dogs' ] },
 {price: 411, sqft: 1314, br: 2, pets: [ 'dogs' ] },
 {price: 275, sqft: 1200, br: 1, pets: [ 'cats' ]},
 {price: 500, sqft: 650, br: 1, pets: [] },
];

. . .
```

```
<html>
<head>
<script src="http://d3js.org/d3.v4.min.js"></script>
</head>

<body>
<svg class="chart" height="900" width="900">
</svg>

<script>

var values = [
 {price: 700, sqft: 3000, br: 3, pets: [ 'cats', 'dogs' ] },
 {price: 445, sqft: 1700, br: 2, pets: [] },
 {price: 421, sqft: 1455, br: 2, pets: [ 'cats', 'dogs' ] },
 {price: 411, sqft: 1314, br: 2, pets: [ 'dogs' ] },
 {price: 275, sqft: 1200, br: 1, pets: [ 'cats' ]},
 {price: 500, sqft: 650, br: 1, pets: [] },
];

. . .
```

```
<html>
<head>
<script src="http://d3js.org/d3.v4.min.js"></script>
</head>

<body>
<svg class="chart" height="900" width="900">
</svg>

<script>

var values = [
 {price: 700, sqft: 3000, br: 3, pets: [ 'cats', 'dogs' ] },
 {price: 445, sqft: 1700, br: 2, pets: [] },
 {price: 421, sqft: 1455, br: 2, pets: [ 'cats', 'dogs' ] },
 {price: 411, sqft: 1314, br: 2, pets: [ 'dogs' ] },
 {price: 275, sqft: 1200, br: 1, pets: [ 'cats' ]},
 {price: 500, sqft: 650, br: 1, pets: [] },
];

. . .
```

```
<html>
<head>
<script src="http://d3js.org/d3.v4.min.js"></script>
</head>

<body>
<svg class="chart" height="900" width="900">
</svg>

<script>

var values = [
 {price: 700, sqft: 3000, br: 3, pets: [ 'cats', 'dogs' ] },
 {price: 445, sqft: 1700, br: 2, pets: [] },
 {price: 421, sqft: 1455, br: 2, pets: [ 'cats', 'dogs' ] },
 {price: 411, sqft: 1314, br: 2, pets: [ 'dogs' ] },
 {price: 275, sqft: 1200, br: 1, pets: [ 'cats' ]},
 {price: 500, sqft: 650, br: 1, pets: [] },
];

. . .
```

```
<html>
<head>
<script src="http://d3js.org/d3.v4.min.js"></script>
</head>

<body>
<svg class="chart" height="900" width="900">
</svg>

<script>

var values = [
 {price: 700, sqft: 3000, br: 3, pets: [ 'cats', 'dogs' ] },
 {price: 445, sqft: 1700, br: 2, pets: [] },
 {price: 421, sqft: 1455, br: 2, pets: [ 'cats', 'dogs' ] },
 {price: 411, sqft: 1314, br: 2, pets: [ 'dogs' ] },
 {price: 275, sqft: 1200, br: 1, pets: [ 'cats' ]},
 {price: 500, sqft: 650, br: 1, pets: [] },
];

. . .
```


d3-homes.html
Guest
d3-homes.html
4,000
3,500
3,000
2,500
2,000
1,500
1,000
500
0
0
100
200
300
400
500
600
700
800

```
var svg = d3.select("svg");
var selection = svg.selectAll("g")
    .data(values)
    .enter()
    .append("g")
    .attr("transform", "translate(10,10)");

selection.append("circle")
  .attr("cx", (d,i) => { return d.price / 2; })
  .attr("cy", (d,i) => { return (4000 -  d.sqft)/(4000/400) ; })
  .attr("r", (d,i) => { return d.br * 10 ; })
  .style("fill", (d,i) => { return color(d.pets); })
  .style("opacity", "0.5")
  .append("svg:title").text( (d,i) => { return print(d); });

function color(pets) {
    var dogs = pets.indexOf('dogs') != -1;
    var cats = pets.indexOf('cats') != -1;
    if (dogs) return cats ? 'purple' : 'blue' ;
    else return cats ? 'red' : 'gray';
}

function print(home) {
  return `$${home.price}k, ${home.sqft}sqft, ${home.br} BRs`;
}
```

```
var svg = d3.select("svg");
var selection = svg.selectAll("g")
    .data(values)
    .enter()
    .append("g")
    .attr("transform", "translate(10,10)");

selection.append("circle")
  .attr("cx", (d,i) => { return d.price / 2; })
  .attr("cy", (d,i) => { return (4000 -  d.sqft)/(4000/400) ; })
  .attr("r", (d,i) => { return d.br * 10 ; })
  .style("fill", (d,i) => { return color(d.pets); })
  .style("opacity", "0.5")
  .append("svg:title").text( (d,i) => { return print(d); });

function color(pets) {
    var dogs = pets.indexOf('dogs') != -1;
    var cats = pets.indexOf('cats') != -1;
    if (dogs) return cats ? 'purple' : 'blue' ;
    else return cats ? 'red' : 'gray';
}

function print(home) {
  return `$${home.price}k, ${home.sqft}sqft, ${home.br} BRs`;
}
```

```
var svg = d3.select("svg");
var selection = svg.selectAll("g")
    .data(values)
    .enter()
    .append("g")
    .attr("transform", "translate(10,10)");

selection.append("circle")
  .attr("cx", (d,i) => { return d.price / 2; })
  .attr("cy", (d,i) => { return (4000 -  d.sqft)/(4000/400) ; })
  .attr("r", (d,i) => { return d.br * 10 ; })
  .style("fill", (d,i) => { return color(d.pets); })
  .style("opacity", "0.5")
  .append("svg:title").text( (d,i) => { return print(d); });

function color(pets) {
    var dogs = pets.indexOf('dogs') != -1;
    var cats = pets.indexOf('cats') != -1;
    if (dogs) return cats ? 'purple' : 'blue' ;
    else return cats ? 'red' : 'gray';
}

function print(home) {
  return `$${home.price}k, ${home.sqft}sqft, ${home.br} BRs`;
}
```

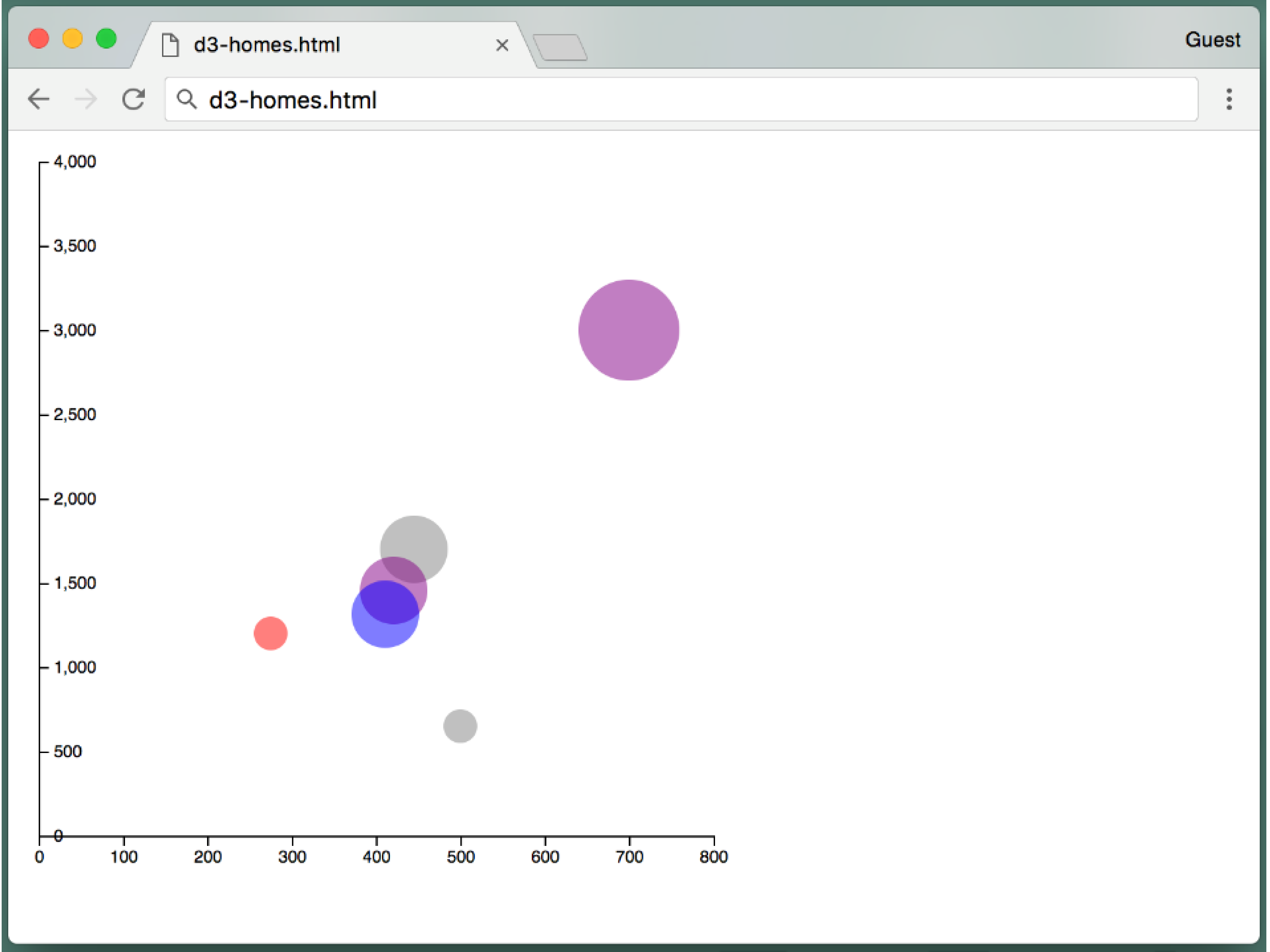
```
var svg = d3.select("svg");
var selection = svg.selectAll("g")
    .data(values)
    .enter()
    .append("g")
    .attr("transform", "translate(10,10)");

selection.append("circle")
  .attr("cx", (d,i) => { return d.price / 2; })
  .attr("cy", (d,i) => { return (4000 -  d.sqft)/(4000/400) ; })
  .attr("r", (d,i) => { return d.br * 10 ; })
  .style("fill", (d,i) => { return color(d.pets); })
  .style("opacity", "0.5")
  .append("svg:title").text( (d,i) => { return print(d); });

function color(pets) {
    var dogs = pets.indexOf('dogs') != -1;
    var cats = pets.indexOf('cats') != -1;
    if (dogs) return cats ? 'purple' : 'blue' ;
    else return cats ? 'red' : 'gray';
}

function print(home) {
  return `$${home.price}k, ${home.sqft}sqft, ${home.br} BRs`;
}
```

```
var svg = d3.select("svg");
var selection = svg.selectAll("g")
    .data(values)
    .enter()
    .append("g")
    .attr("transform", "translate(10,10)");

selection.append("circle")
  .attr("cx", (d,i) => { return d.price / 2; })
  .attr("cy", (d,i) => { return (4000 -  d.sqft)/(4000/400) ; })
  .attr("r", (d,i) => { return d.br * 10 ; })
  .style("fill", (d,i) => { return color(d.pets); })
  .style("opacity", "0.5")
  .append("svg:title").text( (d,i) => { return print(d); });

function color(pets) {
    var dogs = pets.indexOf('dogs') != -1;
    var cats = pets.indexOf('cats') != -1;
    if (dogs) return cats ? 'purple' : 'blue' ;
    else return cats ? 'red' : 'gray';
}

function print(home) {
  return `$${home.price}k, ${home.sqft}sqft, ${home.br} BRs`;
}
```

```
var svg = d3.select("svg");
var selection = svg.selectAll("g")
    .data(values)
    .enter()
    .append("g")
    .attr("transform", "translate(10,10)");

selection.append("circle")
  .attr("cx", (d,i) => { return d.price / 2; })
  .attr("cy", (d,i) => { return (4000 -  d.sqft)/(4000/400) ; })
  .attr("r", (d,i) => { return d.br * 10 ; })
  .style("fill", (d,i) => { return color(d.pets); })
  .style("opacity", "0.5")
  .append("svg:title").text( (d,i) => { return print(d); });

function color(pets) {
    var dogs = pets.indexOf('dogs') != -1;
    var cats = pets.indexOf('cats') != -1;
    if (dogs) return cats ? 'purple' : 'blue' ;
    else return cats ? 'red' : 'gray';
}

function print(home) {
  return `$${home.price}k, ${home.sqft}sqft, ${home.br} BRs`;
}
```

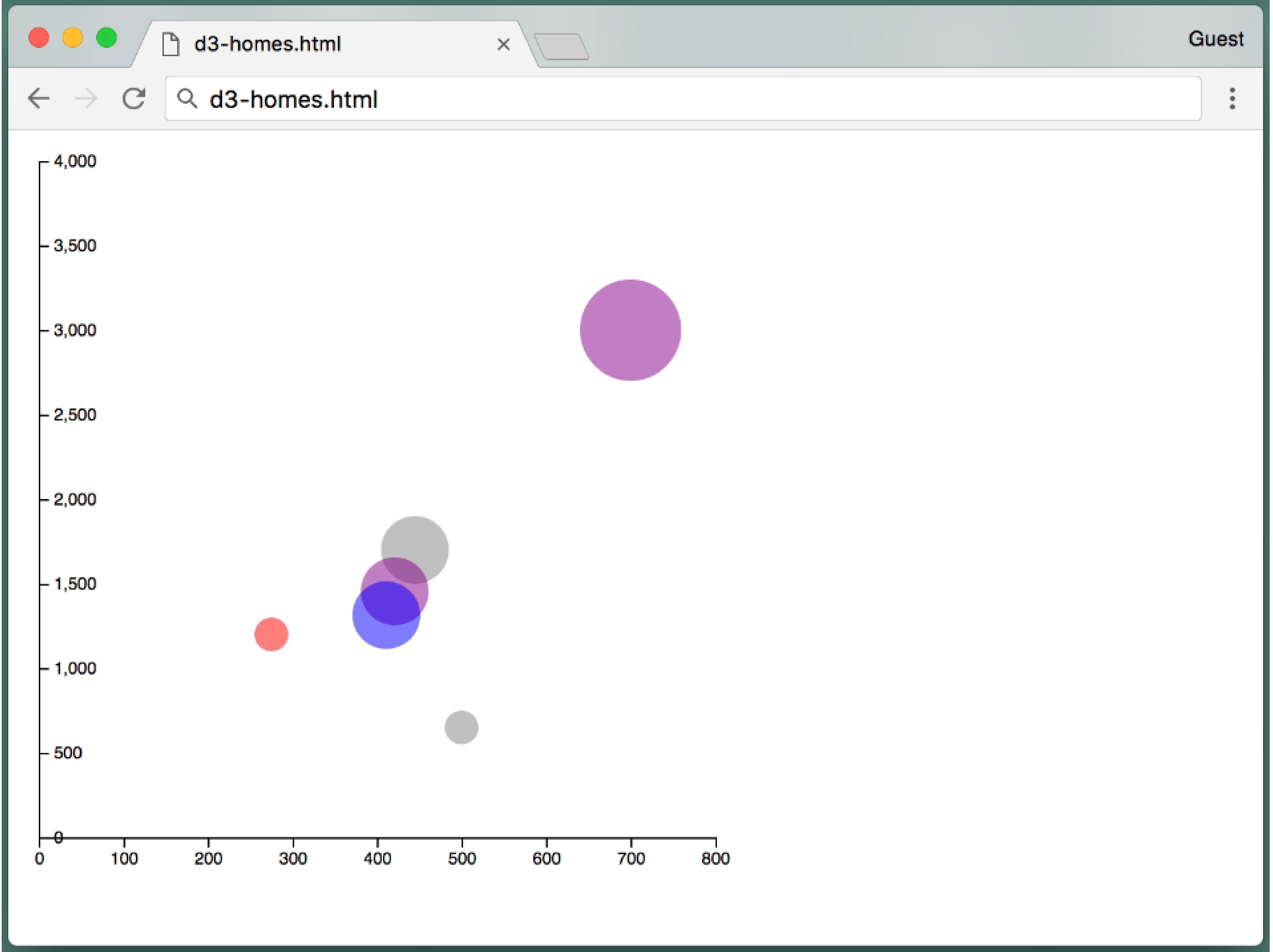
```
var svg = d3.select("svg");
var selection = svg.selectAll("g")
    .data(values)
    .enter()
    .append("g")
    .attr("transform", "translate(10,10)");

selection.append("circle")
  .attr("cx", (d,i) => { return d.price / 2; })
  .attr("cy", (d,i) => { return (4000 -  d.sqft)/(4000/400) ; })
  .attr("r", (d,i) => { return d.br * 10 ; })
  .style("fill", (d,i) => { return color(d.pets); })
  .style("opacity", "0.5")
  .append("svg:title").text( (d,i) => { return print(d); });

function color(pets) {
    var dogs = pets.indexOf('dogs') != -1;
    var cats = pets.indexOf('cats') != -1;
    if (dogs) return cats ? 'purple' : 'blue' ;
    else return cats ? 'red' : 'gray';
}

function print(home) {
  return `$${home.price}k, ${home.sqft}sqft, ${home.br} BRs`;
}
```

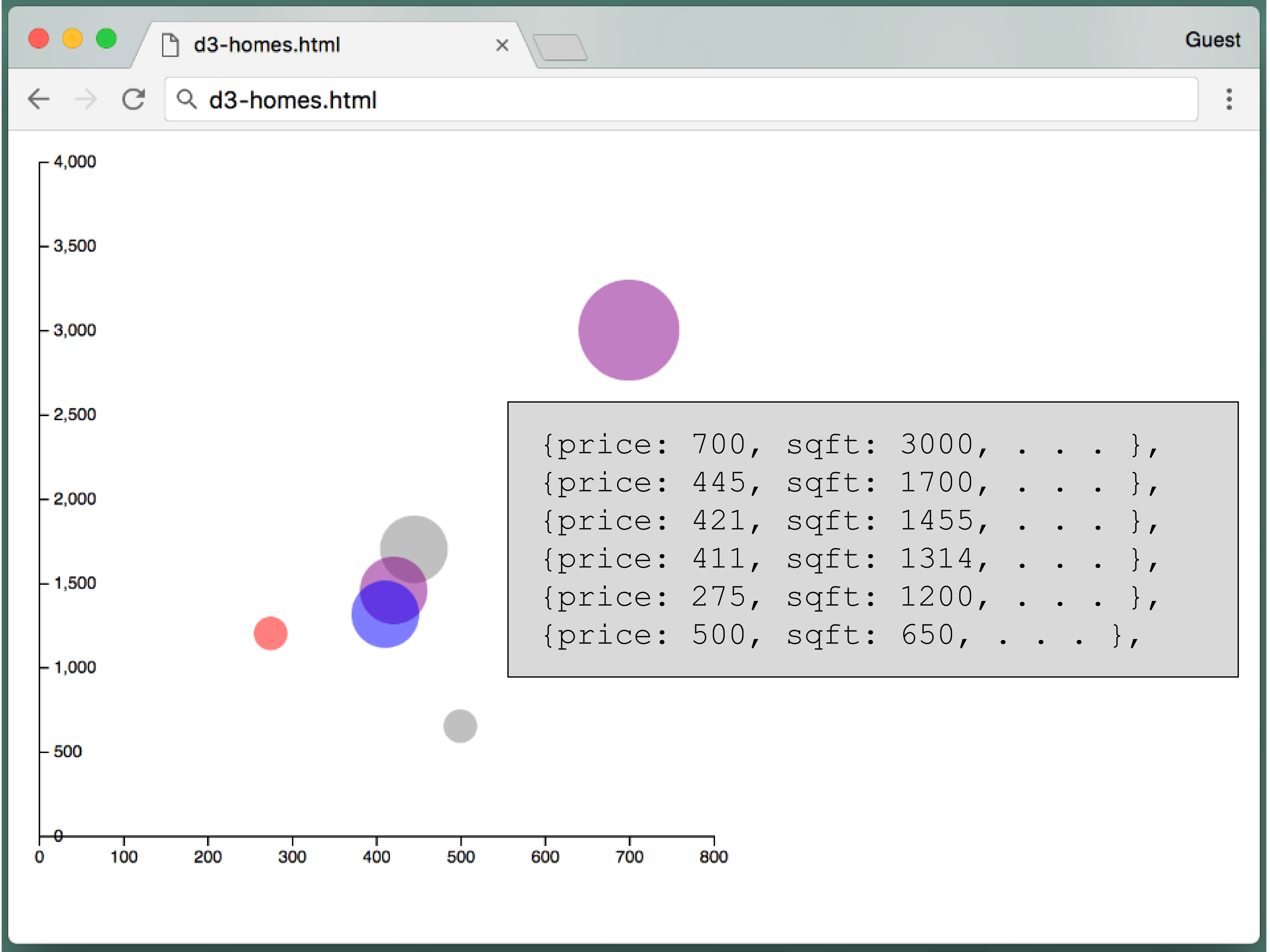
```
var svg = d3.select("svg");
var selection = svg.selectAll("g")
    .data(values)
    .enter()
    .append("g")
    .attr("transform", "translate(10,10)");

selection.append("circle")
  .attr("cx", (d,i) => { return d.price / 2; })
  .attr("cy", (d,i) => { return (4000 -  d.sqft)/(4000/400) ; })
  .attr("r", (d,i) => { return d.br * 10 ; })
  .style("fill", (d,i) => { return color(d.pets); })
  .style("opacity", "0.5")
  .append("svg:title").text( (d,i) => { return print(d); });

function color(pets) {
    var dogs = pets.indexOf('dogs') != -1;
    var cats = pets.indexOf('cats') != -1;
    if (dogs) return cats ? 'purple' : 'blue' ;
    else return cats ? 'red' : 'gray';
}

function print(home) {
  return `$${home.price}k, ${home.sqft}sqft, ${home.br} BRs`;
}
```


d3-homes.html
Guest
d3-homes.html
4,000
3,500
3,000
2,500
2,000
1,500
1,000
500
0
0
100
200
300
400
500
600
700
800

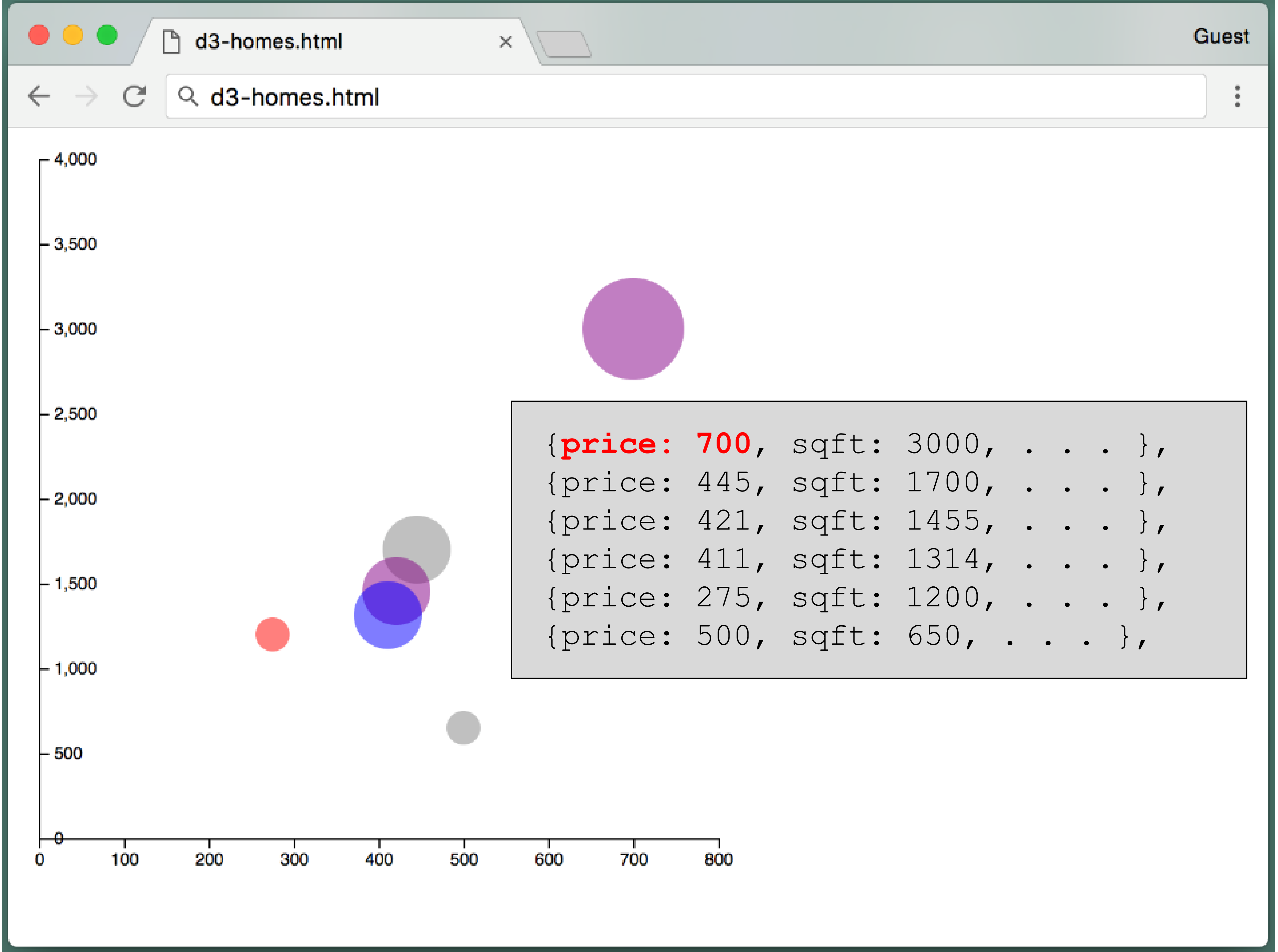
```
var svg = d3.select("svg");
var selection = svg.selectAll("g")
    .data(values)
    .enter()
    .append("g")
    .attr("transform", "translate(10,10)");

selection.append("circle")
  .attr("cx", (d,i) => { return d.price / 2; })
  .attr("cy", (d,i) => { return (4000 -  d.sqft)/(4000/400) ; })
  .attr("r", (d,i) => { return d.br * 10 ; })
  .style("fill", (d,i) => { return color(d.pets); })
  .style("opacity", "0.5")
  .append("svg:title").text( (d,i) => { return print(d); });

function color(pets) {
    var dogs = pets.indexOf('dogs') != -1;
    var cats = pets.indexOf('cats') != -1;
    if (dogs) return cats ? 'purple' : 'blue' ;
    else return cats ? 'red' : 'gray';
}

function print(home) {
  return `$${home.price}k, ${home.sqft}sqft, ${home.br} BRs`;
}
```

```
var svg = d3.select("svg");
var selection = svg.selectAll("g")
    .data(values)
    .enter()
    .append("g")
    .attr("transform", "translate(10,10)");

selection.append("circle")
  .attr("cx", (d,i) => { return d.price / 2; })
  .attr("cy", (d,i) => { return (4000 -  d.sqft)/(4000/400) ; })
  .attr("r", (d,i) => { return d.br * 10 ; })
  .style("fill", (d,i) => { return color(d.pets); })
  .style("opacity", "0.5")
  .append("svg:title").text( (d,i) => { return print(d); });

function color(pets) {
    var dogs = pets.indexOf('dogs') != -1;
    var cats = pets.indexOf('cats') != -1;
    if (dogs) return cats ? 'purple' : 'blue' ;
    else return cats ? 'red' : 'gray';
}

function print(home) {
  return `$${home.price}k, ${home.sqft}sqft, ${home.br} BRs`;
}
```

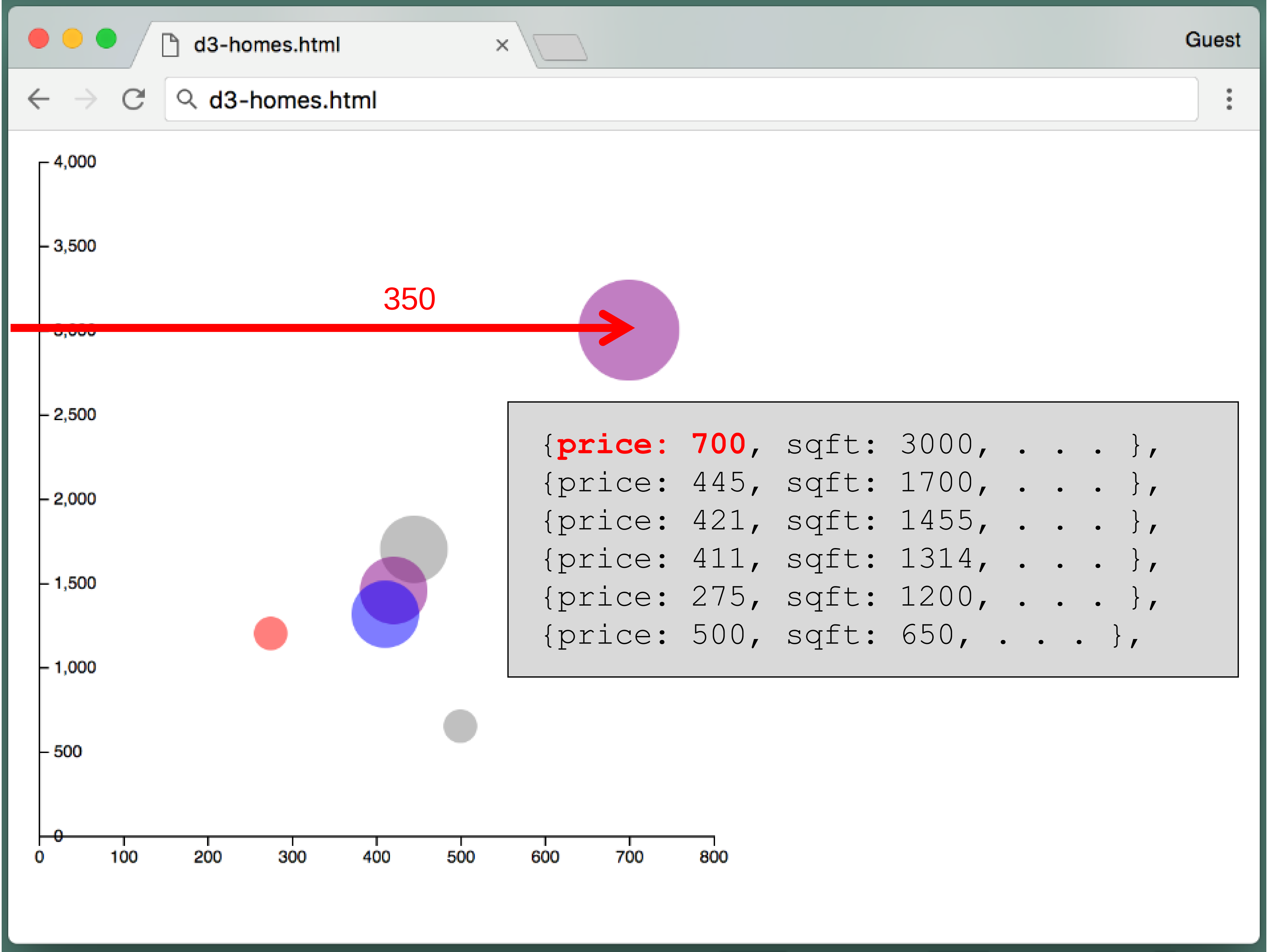

d3-homes.html
Guest
d3-homes.html
4,000
3,500
3,000
2,500
2,000
1,500
1,000
500
0
0
100
200
300
400
500
600
700
800



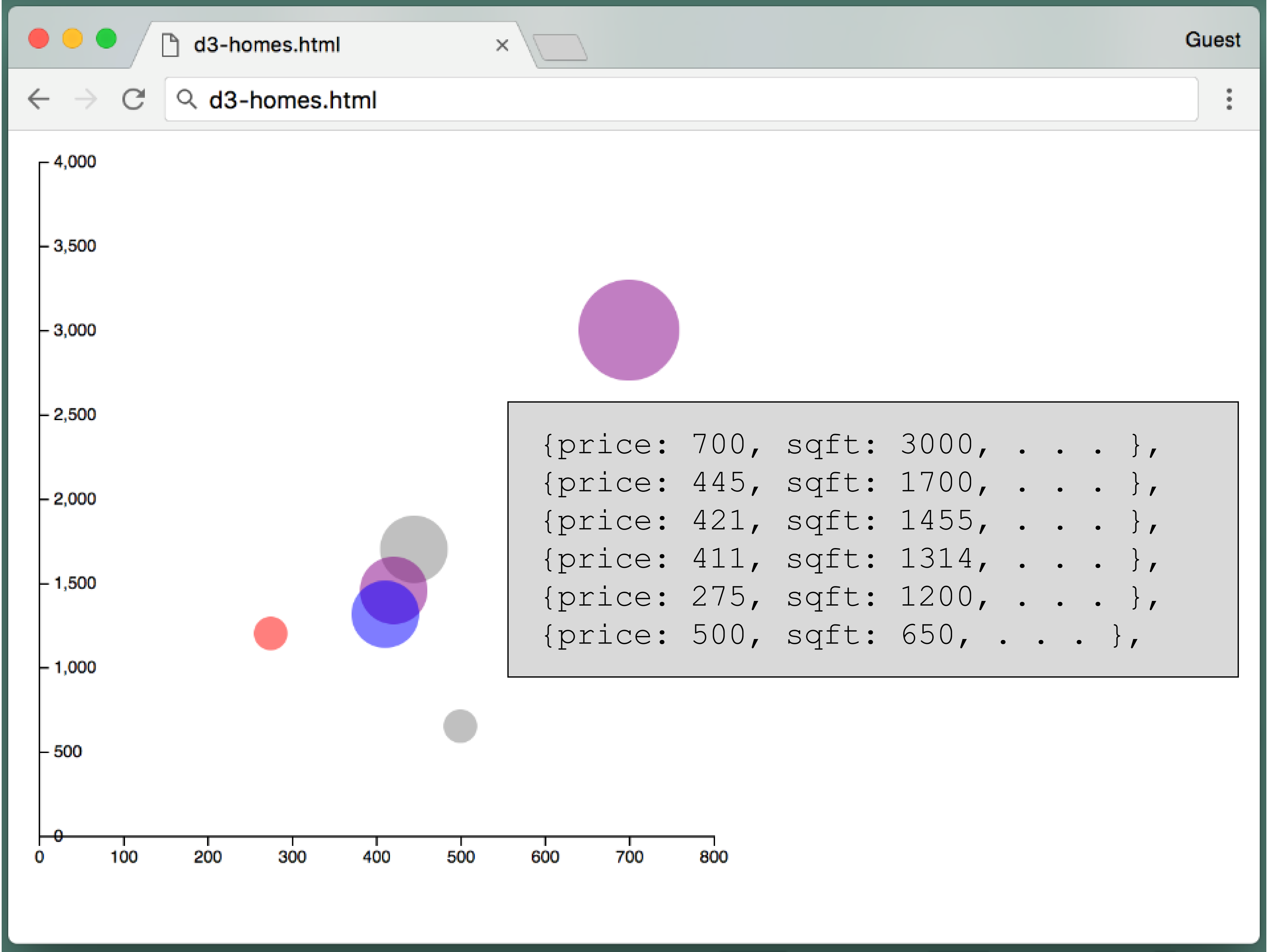
```
{price: 700, sqft: 3000, . . . },
{price: 445, sqft: 1700, . . . },
{price: 421, sqft: 1455, . . . },
{price: 411, sqft: 1314, . . . },
{price: 275, sqft: 1200, . . . },
{price: 500, sqft: 650, . . . },
```



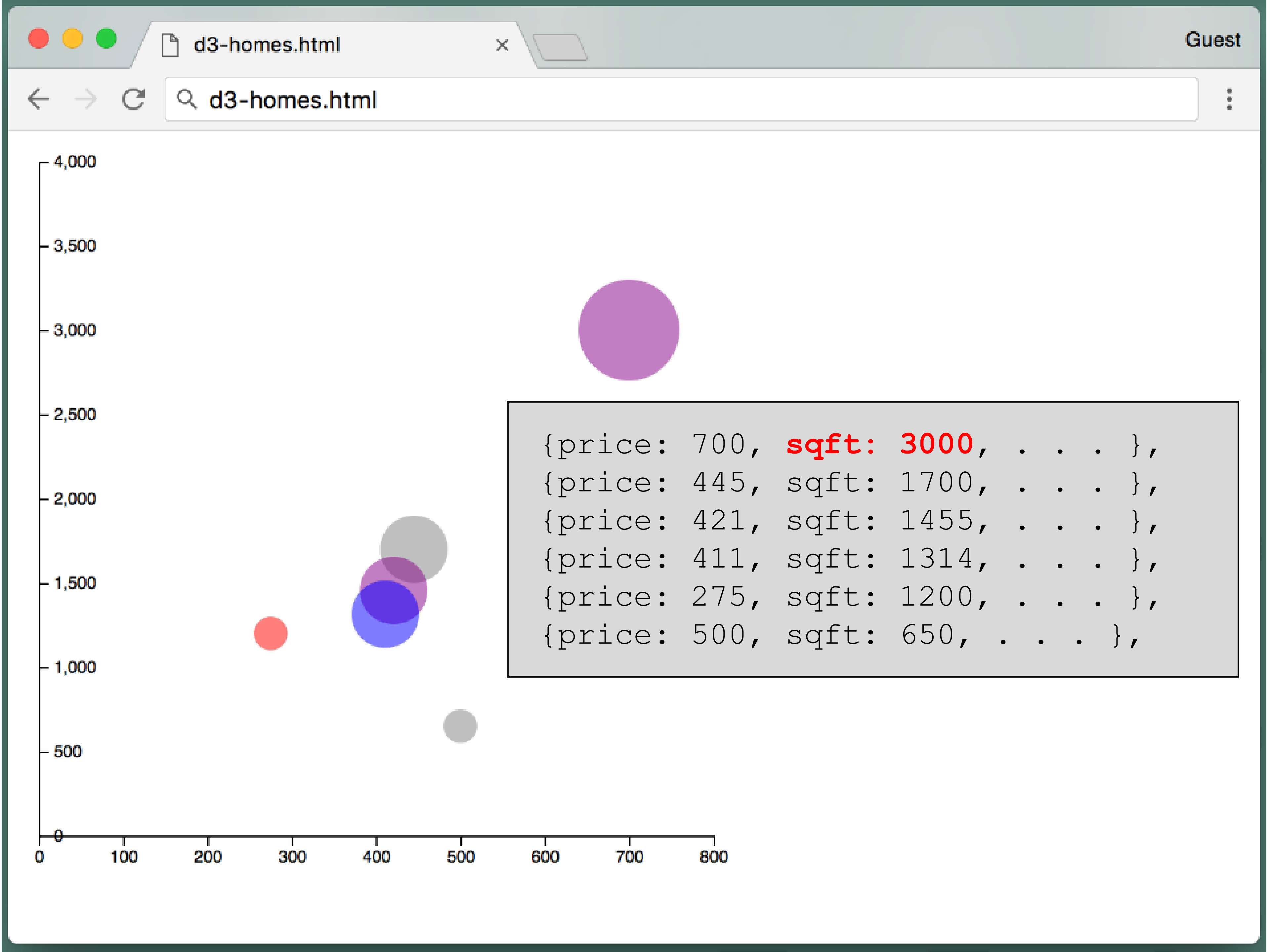
```
{price: 700, sqft: 3000, . . . },
{price: 445, sqft: 1700, . . . },
{price: 421, sqft: 1455, . . . },
{price: 411, sqft: 1314, . . . },
{price: 275, sqft: 1200, . . . },
{price: 500, sqft: 650, . . . },
```



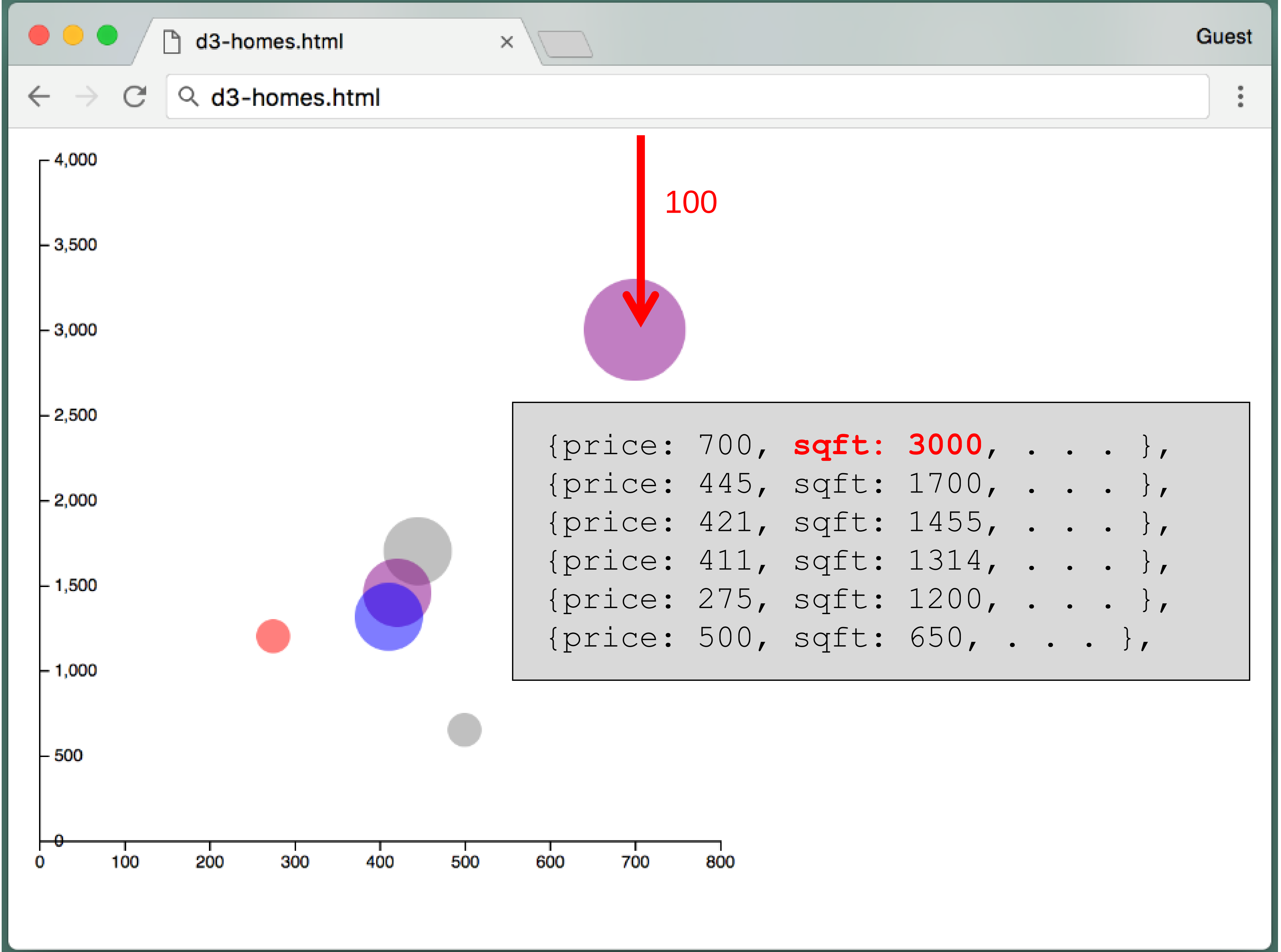
```
{price: 700, sqft: 3000, . . . },
{price: 445, sqft: 1700, . . . },
{price: 421, sqft: 1455, . . . },
{price: 411, sqft: 1314, . . . },
{price: 275, sqft: 1200, . . . },
{price: 500, sqft: 650, . . . },
var svg = d3.select("svg");
var selection = svg.selectAll("g")
    .data(values)
    .enter()
    .append("g")
    .attr("transform", "translate(10,10)");

selection.append("circle")
  .attr("cx", (d,i) => { return d.price / 2; })
  .attr("cy", (d,i) => { return (4000 -  d.sqft)/(4000/400) ; })
  .attr("r", (d,i) => { return d.br * 10 ; })
  .style("fill", (d,i) => { return color(d.pets); })
  .style("opacity", "0.5")
  .append("svg:title").text( (d,i) => { return print(d); });

function color(pets) {
    var dogs = pets.indexOf('dogs') != -1;
    var cats = pets.indexOf('cats') != -1;
    if (dogs) return cats ? 'purple' : 'blue' ;
    else return cats ? 'red' : 'gray';
}

function print(home) {
  return `$${home.price}k, ${home.sqft}sqft, ${home.br} BRs`;
}
```



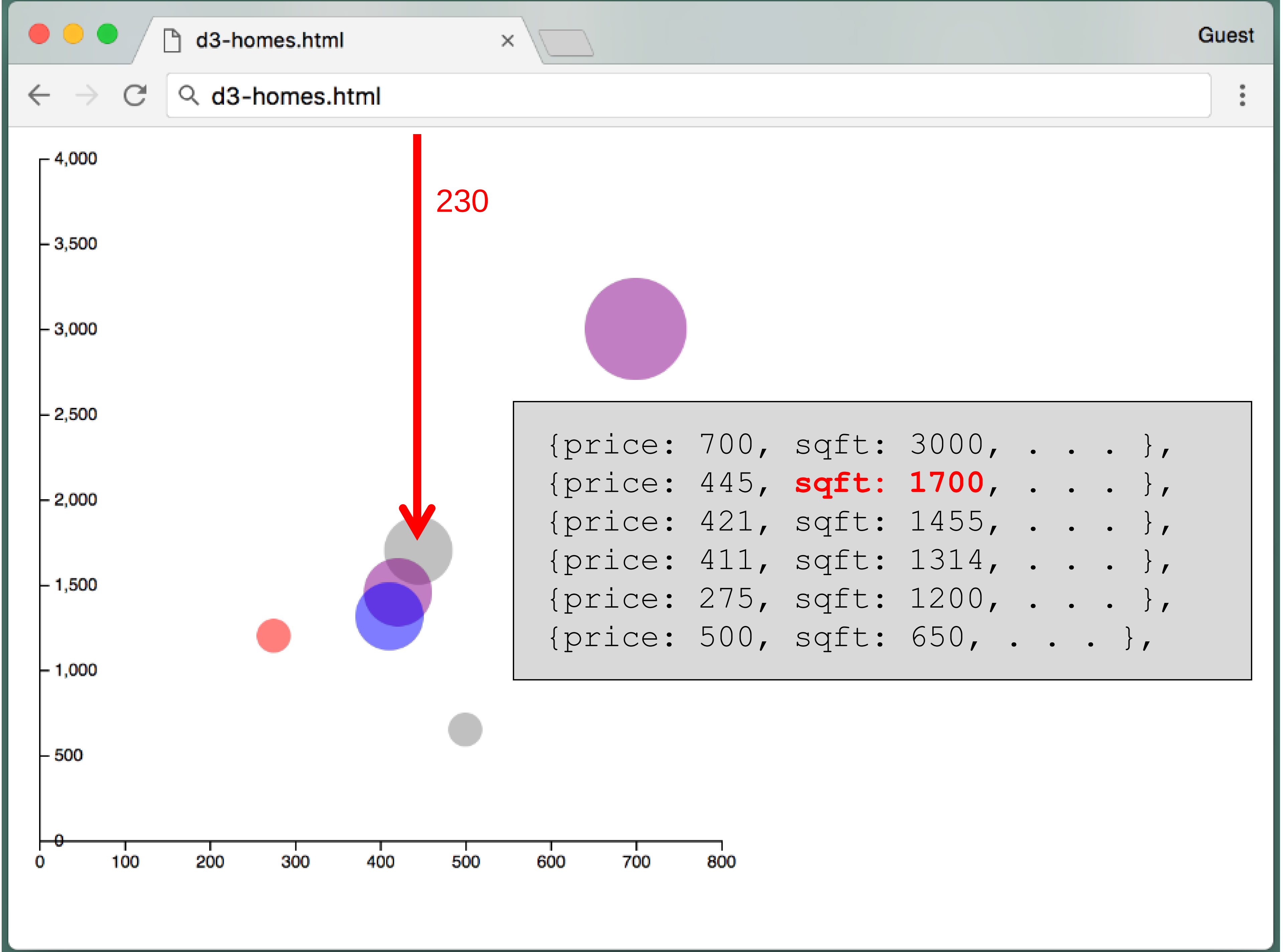
```
{price: 700, sqft: 3000, . . . },
{price: 445, sqft: 1700, . . . },
{price: 421, sqft: 1455, . . . },
{price: 411, sqft: 1314, . . . },
{price: 275, sqft: 1200, . . . },
{price: 500, sqft: 650, . . . },
```



```
{price: 700, sqft: 3000, . . . },
{price: 445, sqft: 1700, . . . },
{price: 421, sqft: 1455, . . . },
{price: 411, sqft: 1314, . . . },
{price: 275, sqft: 1200, . . . },
{price: 500, sqft: 650, . . . },
```



```
{price: 700, sqft: 3000, . . . },
{price: 445, sqft: 1700, . . . },
{price: 421, sqft: 1455, . . . },
{price: 411, sqft: 1314, . . . },
{price: 275, sqft: 1200, . . . },
{price: 500, sqft: 650, . . . },
```



```
{price: 700, sqft: 3000, . . . },
{price: 445, sqft: 1700, . . . },
{price: 421, sqft: 1455, . . . },
{price: 411, sqft: 1314, . . . },
{price: 275, sqft: 1200, . . . },
{price: 500, sqft: 650, . . . },
var svg = d3.select("svg");
var selection = svg.selectAll("g")
    .data(values)
    .enter()
    .append("g")
    .attr("transform", "translate(10,10)");

selection.append("circle")
  .attr("cx", (d,i) => { return d.price / 2; })
  .attr("cy", (d,i) => { return (4000 -  d.sqft)/(4000/400) ; })
  .attr("r", (d,i) => { return d.br * 10 ; })
  .style("fill", (d,i) => { return color(d.pets); })
  .style("opacity", "0.5")
  .append("svg:title").text( (d,i) => { return print(d); });

function color(pets) {
    var dogs = pets.indexOf('dogs') != -1;
    var cats = pets.indexOf('cats') != -1;
    if (dogs) return cats ? 'purple' : 'blue' ;
    else return cats ? 'red' : 'gray';
}

function print(home) {
  return `$${home.price}k, ${home.sqft}sqft, ${home.br} BRs`;
}
```

```
var svg = d3.select("svg");
var selection = svg.selectAll("g")
    .data(values)
    .enter()
    .append("g")
    .attr("transform", "translate(10,10)");

selection.append("circle")
  .attr("cx", (d,i) => { return d.price / 2; })
  .attr("cy", (d,i) => { return (4000 -  d.sqft)/(4000/400) ; })
  .attr("r", (d,i) => { return d.br * 10 ; })
  .style("fill", (d,i) => { return color(d.pets); })
  .style("opacity", "0.5")
  .append("svg:title").text( (d,i) => { return print(d); });

function color(pets) {
    var dogs = pets.indexOf('dogs') != -1;
    var cats = pets.indexOf('cats') != -1;
    if (dogs) return cats ? 'purple' : 'blue' ;
    else return cats ? 'red' : 'gray';
}

function print(home) {
  return `$${home.price}k, ${home.sqft}sqft, ${home.br} BRs`;
}
```

```
var svg = d3.select("svg");
var selection = svg.selectAll("g")
    .data(values)
    .enter()
    .append("g")
    .attr("transform", "translate(10,10)");

selection.append("circle")
  .attr("cx", (d,i) => { return d.price / 2; })
  .attr("cy", (d,i) => { return (4000 -  d.sqft)/(4000/400) ; })
  .attr("r", (d,i) => { return d.br * 10 ; })
  .style("fill", (d,i) => { return color(d.pets); })
  .style("opacity", "0.5")
  .append("svg:title").text( (d,i) => { return print(d); });

function color(pets) {
    var dogs = pets.indexOf('dogs') != -1;
    var cats = pets.indexOf('cats') != -1;
    if (dogs) return cats ? 'purple' : 'blue' ;
    else return cats ? 'red' : 'gray';
}

function print(home) {
  return `$${home.price}k, ${home.sqft}sqft, ${home.br} BRs`;
}
```

```
var svg = d3.select("svg");
var selection = svg.selectAll("g")
    .data(values)
    .enter()
    .append("g")
    .attr("transform", "translate(10,10)");

selection.append("circle")
  .attr("cx", (d,i) => { return d.price / 2; })
  .attr("cy", (d,i) => { return (4000 -  d.sqft)/(4000/400) ; })
  .attr("r", (d,i) => { return d.br * 10 ; })
  .style("fill", (d,i) => { return color(d.pets); })
  .style("opacity", "0.5")
  .append("svg:title").text( (d,i) => { return print(d); });

function color(pets) {
    var dogs = pets.indexOf('dogs') != -1;
    var cats = pets.indexOf('cats') != -1;
    if (dogs) return cats ? 'purple' : 'blue' ;
    else return cats ? 'red' : 'gray';
}

function print(home) {
  return `$${home.price}k, ${home.sqft}sqft, ${home.br} BRs`;
}
```

```
var svg = d3.select("svg");
var selection = svg.selectAll("g")
    .data(values)
    .enter()
    .append("g")
    .attr("transform", "translate(10,10)");

selection.append("circle")
  .attr("cx", (d,i) => { return d.price / 2; })
  .attr("cy", (d,i) => { return (4000 -  d.sqft)/(4000/400) ; })
  .attr("r", (d,i) => { return d.br * 10 ; })
  .style("fill", (d,i) => { return color(d.pets); })
  .style("opacity", "0.5")
  .append("svg:title").text( (d,i) => { return print(d); });

function color(pets) {
    var dogs = pets.indexOf('dogs') != -1;
    var cats = pets.indexOf('cats') != -1;
    if (dogs) return cats ? 'purple' : 'blue' ;
    else return cats ? 'red' : 'gray';
}

function print(home) {
  return `$${home.price}k, ${home.sqft}sqft, ${home.br} BRs`;
}
```

```
var svg = d3.select("svg");
var selection = svg.selectAll("g")
    .data(values)
    .enter()
    .append("g")
    .attr("transform", "translate(10,10)");

selection.append("circle")
  .attr("cx", (d,i) => { return d.price / 2; })
  .attr("cy", (d,i) => { return (4000 -  d.sqft)/(4000/400) ; })
  .attr("r", (d,i) => { return d.br * 10 ; })
  .style("fill", (d,i) => { return color(d.pets); })
  .style("opacity", "0.5")
  .append("svg:title").text( (d,i) => { return print(d); });

function color(pets) {
    var dogs = pets.indexOf('dogs') != -1;
    var cats = pets.indexOf('cats') != -1;
    if (dogs) return cats ? 'purple' : 'blue' ;
    else return cats ? 'red' : 'gray';
}

function print(home) {
  return `$${home.price}k, ${home.sqft}sqft, ${home.br} BRs`;
}
```

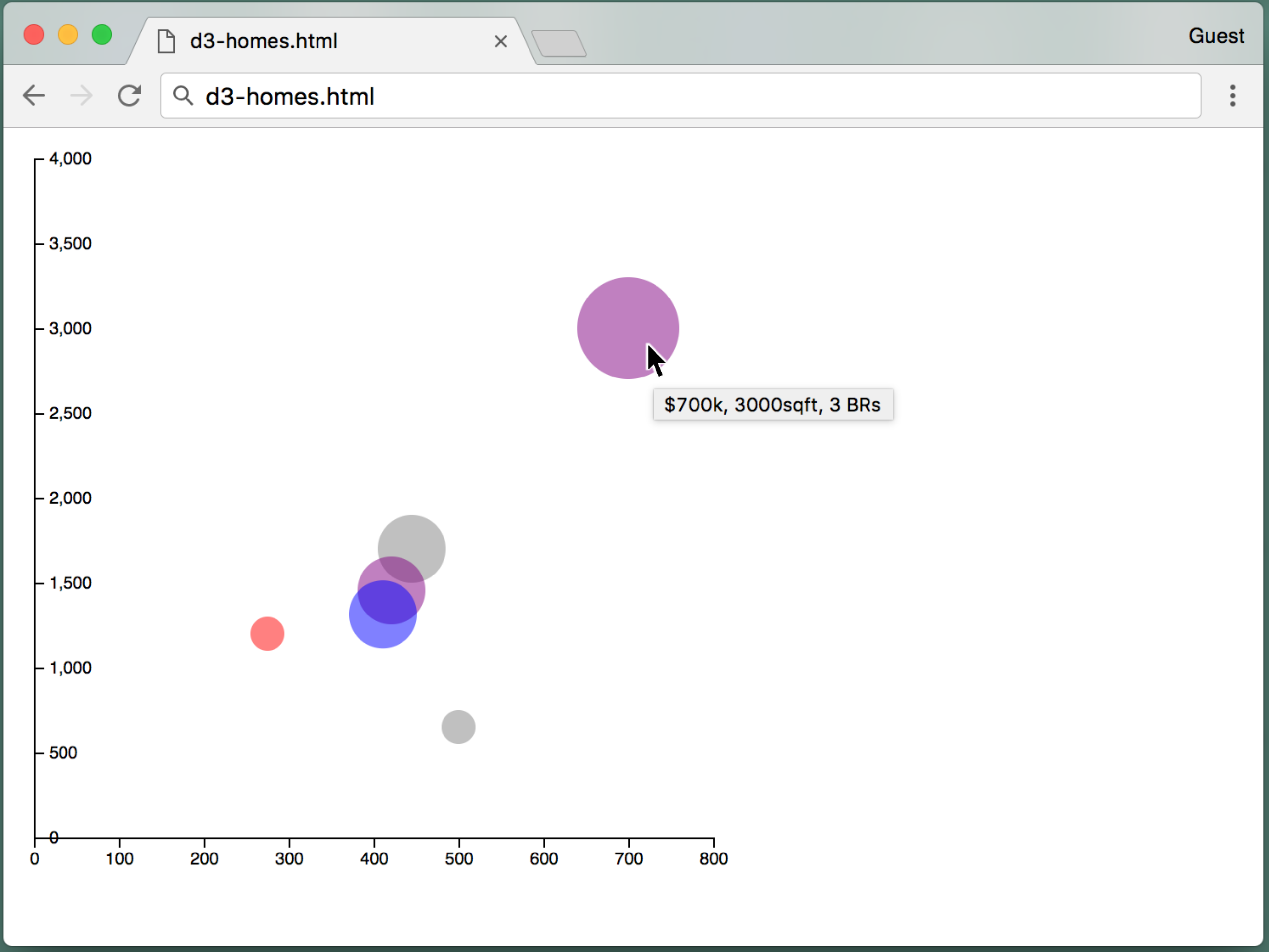
```
var svg = d3.select("svg");
var selection = svg.selectAll("g")
    .data(values)
    .enter()
    .append("g")
    .attr("transform", "translate(10,10)");

selection.append("circle")
  .attr("cx", (d,i) => { return d.price / 2; })
  .attr("cy", (d,i) => { return (4000 -  d.sqft)/(4000/400) ; })
  .attr("r", (d,i) => { return d.br * 10 ; })
  .style("fill", (d,i) => { return color(d.pets); })
  .style("opacity", "0.5")
  .append("svg:title").text( (d,i) => { return print(d); });

function color(pets) {
    var dogs = pets.indexOf('dogs') != -1;
    var cats = pets.indexOf('cats') != -1;
    if (dogs) return cats ? 'purple' : 'blue' ;
    else return cats ? 'red' : 'gray';
}

function print(home) {
  return `$${home.price}k, ${home.sqft}sqft, ${home.br} BRs`;
}
```

```
var svg = d3.select("svg");
var selection = svg.selectAll("g")
    .data(values)
    .enter()
    .append("g")
    .attr("transform", "translate(10,10)");

selection.append("circle")
  .attr("cx", (d,i) => { return d.price / 2; })
  .attr("cy", (d,i) => { return (4000 -  d.sqft)/(4000/400) ; })
  .attr("r", (d,i) => { return d.br * 10 ; })
  .style("fill", (d,i) => { return color(d.pets); })
  .style("opacity", "0.5")
  .append("svg:title").text( (d,i) => { return print(d); });

function color(pets) {
    var dogs = pets.indexOf('dogs') != -1;
    var cats = pets.indexOf('cats') != -1;
    if (dogs) return cats ? 'purple' : 'blue' ;
    else return cats ? 'red' : 'gray';
}

function print(home) {
  return `$${home.price}k, ${home.sqft}sqft, ${home.br} BRs`;
}
```

```
var svg = d3.select("svg");
var selection = svg.selectAll("g")
    .data(values)
    .enter()
    .append("g")
    .attr("transform", "translate(10,10)");

selection.append("circle")
  .attr("cx", (d,i) => { return d.price / 2; })
  .attr("cy", (d,i) => { return (4000 -  d.sqft)/(4000/400) ; })
  .attr("r", (d,i) => { return d.br * 10 ; })
  .style("fill", (d,i) => { return color(d.pets); })
  .style("opacity", "0.5")
  .append("svg:title").text( (d,i) => { return print(d); });

function color(pets) {
    var dogs = pets.indexOf('dogs') != -1;
    var cats = pets.indexOf('cats') != -1;
    if (dogs) return cats ? 'purple' : 'blue' ;
    else return cats ? 'red' : 'gray';
}

function print(home) {
  return `$${home.price}k, ${home.sqft}sqft, ${home.br} BRs`;
}
```

```
var svg = d3.select("svg");
var selection = svg.selectAll("g")
    .data(values)
    .enter()
    .append("g")
    .attr("transform", "translate(10,10)");

selection.append("circle")
  .attr("cx", (d,i) => { return d.price / 2; })
  .attr("cy", (d,i) => { return (4000 -  d.sqft)/(4000/400) ; })
  .attr("r", (d,i) => { return d.br * 10 ; })
  .style("fill", (d,i) => { return color(d.pets); })
  .style("opacity", "0.5")
  .append("svg:title").text( (d,i) => { return print(d); });

function color(pets) {
    var dogs = pets.indexOf('dogs') != -1;
    var cats = pets.indexOf('cats') != -1;
    if (dogs) return cats ? 'purple' : 'blue' ;
    else return cats ? 'red' : 'gray';
}

function print(home) {
  return `$${home.price}k, ${home.sqft}sqft, ${home.br} BRs`;
}
```

```
var svg = d3.select("svg");
var selection = svg.selectAll("g")
    .data(values)
    .enter()
    .append("g")
    .attr("transform", "translate(10,10)");

selection.append("circle")
  .attr("cx", (d,i) => { return d.price / 2; })
  .attr("cy", (d,i) => { return (4000 -  d.sqft)/(4000/400) ; })
  .attr("r", (d,i) => { return d.br * 10 ; })
  .style("fill", (d,i) => { return color(d.pets); })
  .style("opacity", "0.5")
  .append("svg:title").text( (d,i) => { return print(d); });

function color(pets) {
    var dogs = pets.indexOf('dogs') != -1;
    var cats = pets.indexOf('cats') != -1;
    if (dogs) return cats ? 'purple' : 'blue' ;
    else return cats ? 'red' : 'gray';
}

function print(home) {
  return `$${home.price}k, ${home.sqft}sqft, ${home.br} BRs`;
}
```

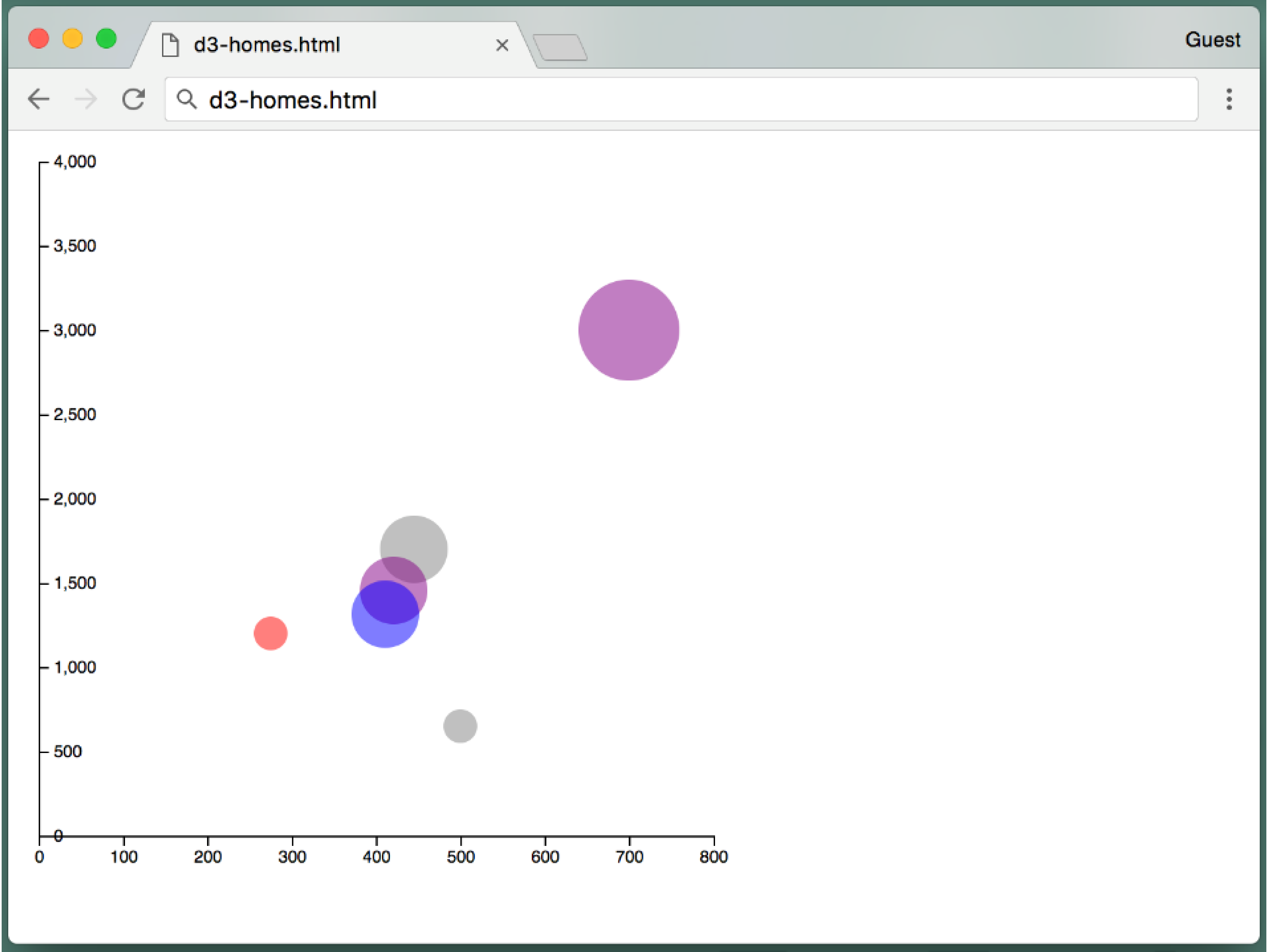


$700k, 3000sqft, 3 BRs

```
var svg = d3.select("svg");
var selection = svg.selectAll("g")
    .data(values)
    .enter()
    .append("g")
    .attr("transform", "translate(10,10)");

selection.append("circle")
  .attr("cx", (d,i) => { return d.price / 2; })
  .attr("cy", (d,i) => { return (4000 -  d.sqft)/(4000/400) ; })
  .attr("r", (d,i) => { return d.br * 10 ; })
  .style("fill", (d,i) => { return color(d.pets); })
  .style("opacity", "0.5")
  .append("svg:title").text( (d,i) => { return print(d); });

function color(pets) {
    var dogs = pets.indexOf('dogs') != -1;
    var cats = pets.indexOf('cats') != -1;
    if (dogs) return cats ? 'purple' : 'blue' ;
    else return cats ? 'red' : 'gray';
}

function print(home) {
  return `$${home.price}k, ${home.sqft}sqft, ${home.br} BRs`;
}
```

```
var svg = d3.select("svg");
var selection = svg.selectAll("g")
    .data(values)
    .enter()
    .append("g")
    .attr("transform", "translate(10,10)");

selection.append("circle")
  .attr("cx", (d,i) => { return d.price / 2; })
  .attr("cy", (d,i) => { return (4000 -  d.sqft)/(4000/400) ; })
  .attr("r", (d,i) => { return d.br * 10 ; })
  .style("fill", (d,i) => { return color(d.pets); })
  .style("opacity", "0.5")
  .append("svg:title").text( (d,i) => { return print(d); });

function color(pets) {
    var dogs = pets.indexOf('dogs') != -1;
    var cats = pets.indexOf('cats') != -1;
    if (dogs) return cats ? 'purple' : 'blue' ;
    else return cats ? 'red' : 'gray';
}

function print(home) {
  return `$${home.price}k, ${home.sqft}sqft, ${home.br} BRs`;
}
```

```
var svg = d3.select("svg");
var selection = svg.selectAll("g")
    .data(values)
    .enter()
    .append("g")
    .attr("transform", "translate(10,10)");

selection.append("circle")
  .attr("cx", (d,i) => { return d.price / 2; })
  .attr("cy", (d,i) => { return (4000 -  d.sqft)/(4000/400) ; })
  .attr("r", (d,i) => { return d.br * 10 ; })
  .style("fill", (d,i) => { return color(d.pets); })
  .style("opacity", "0.5")
  .append("svg:title").text( (d,i) => { return print(d); });

function color(pets) {
    var dogs = pets.indexOf('dogs') != -1;
    var cats = pets.indexOf('cats') != -1;
    if (dogs) return cats ? 'purple' : 'blue' ;
    else return cats ? 'red' : 'gray';
}

function print(home) {
  return `$${home.price}k, ${home.sqft}sqft, ${home.br} BRs`;
}
```


d3-homes.html
Guest
d3-homes.html
4,000
3,500
3,000
2,500
2,000
1,500
1,000
500
0
0
100
200
300
400
500
600
700
800

```
var width = 400;
var height = 400;

// draw the x-axis
var xScale = d3.scaleLinear()
    .domain([0, width*2])
    .range([0, width]);
var xAxis = d3.axisBottom(xScale);

svg.append("g")
    .attr("transform", "translate(10,410)")
    .call(xAxis);

// draw the y-axis
var yScale = d3.scaleLinear()
    .range([height,0]);
    .domain([0, 4000]);
var yAxis = d3.axisRight(yScale);

svg.append("g").attr("transform", "translate(10, 10)")
    .call(yAxis);
```

```
var width = 400;
var height = 400;

// draw the x-axis
var xScale = d3.scaleLinear()
    .domain([0, width*2])
    .range([0, width]);
var xAxis = d3.axisBottom(xScale);

svg.append("g")
    .attr("transform", "translate(10,410)")
    .call(xAxis);

// draw the y-axis
var yScale = d3.scaleLinear()
    .range([height,0]);
    .domain([0, 4000]);
var yAxis = d3.axisRight(yScale);

svg.append("g").attr("transform", "translate(10, 10)")
    .call(yAxis);
```

```
var width = 400;
var height = 400;

// draw the x-axis
var xScale = d3.scaleLinear()
    .domain([0, width*2])
    .range([0, width]);
var xAxis = d3.axisBottom(xScale);

svg.append("g")
    .attr("transform", "translate(10,410)")
    .call(xAxis);

// draw the y-axis
var yScale = d3.scaleLinear()
    .range([height,0]);
    .domain([0, 4000]);
var yAxis = d3.axisRight(yScale);

svg.append("g").attr("transform", "translate(10, 10)")
    .call(yAxis);
```

```
var width = 400;
var height = 400;

// draw the x-axis
var xScale = d3.scaleLinear()
    .domain([0, width*2])
    .range([0, width]);
var xAxis = d3.axisBottom(xScale);

svg.append("g")
    .attr("transform", "translate(10,410)")
    .call(xAxis);

// draw the y-axis
var yScale = d3.scaleLinear()
    .range([height,0]);
    .domain([0, 4000]);
var yAxis = d3.axisRight(yScale);

svg.append("g").attr("transform", "translate(10, 10)")
    .call(yAxis);
```

```
var width = 400;
var height = 400;

// draw the x-axis
var xScale = d3.scaleLinear()
    .domain([0, width*2])
    .range([0, width]);
var xAxis = d3.axisBottom(xScale);

svg.append("g")
    .attr("transform", "translate(10,410)")
    .call(xAxis);

// draw the y-axis
var yScale = d3.scaleLinear()
    .range([height,0]);
    .domain([0, 4000]);
var yAxis = d3.axisRight(yScale);

svg.append("g").attr("transform", "translate(10, 10)")
    .call(yAxis);
```

```
var width = 400;
var height = 400;

// draw the x-axis
var xScale = d3.scaleLinear()
    .domain([0, width*2])
    .range([0, width]);
var xAxis = d3.axisBottom(xScale);

svg.append("g")
    .attr("transform", "translate(10,410)")
    .call(xAxis);

// draw the y-axis
var yScale = d3.scaleLinear()
    .range([height,0]);
    .domain([0, 4000]);
var yAxis = d3.axisRight(yScale);

svg.append("g").attr("transform", "translate(10, 10)")
    .call(yAxis);
```

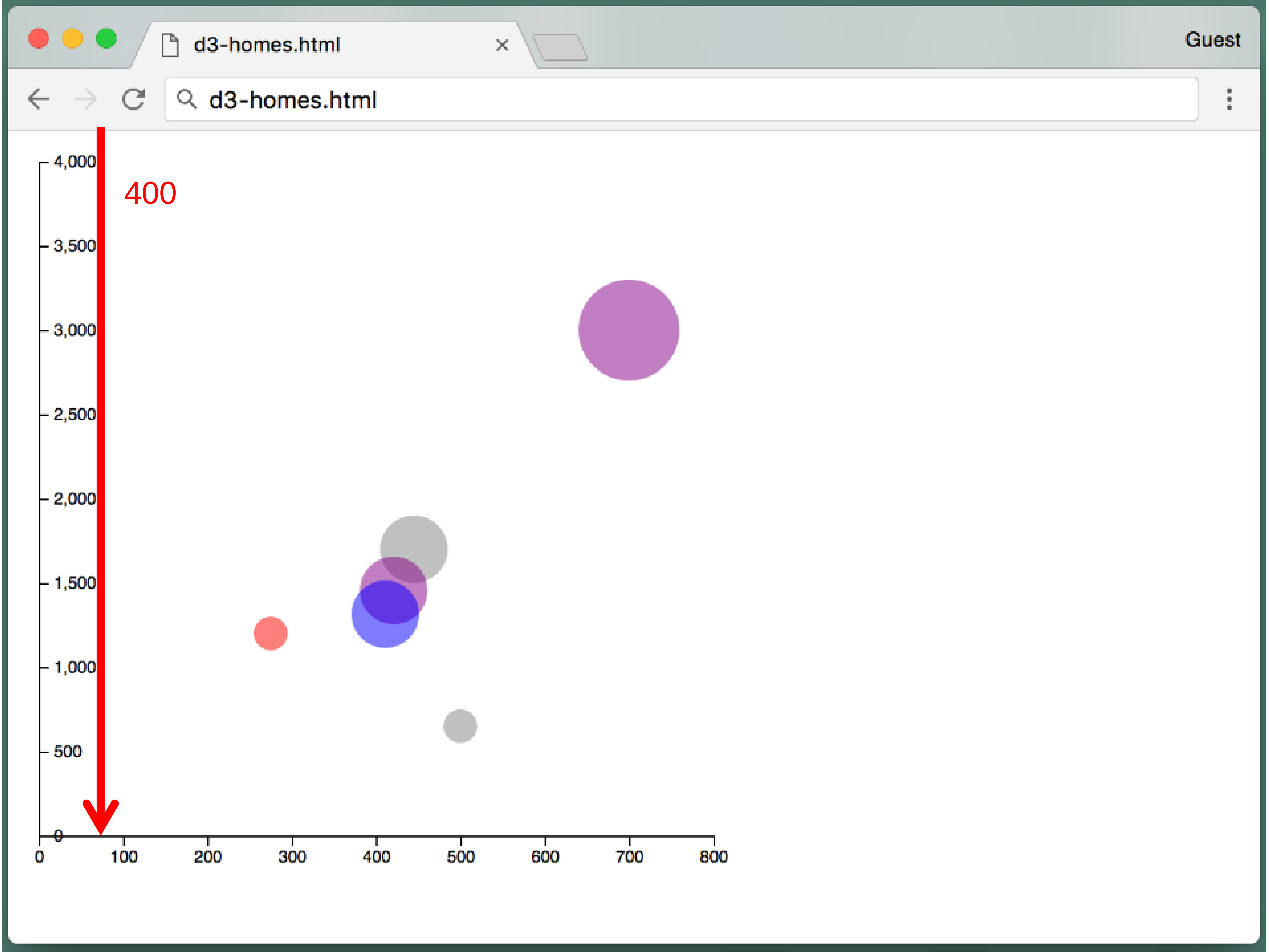
```
var width = 400;
var height = 400;

// draw the x-axis
var xScale = d3.scaleLinear()
    .domain([0, width*2])
    .range([0, width]);
var xAxis = d3.axisBottom(xScale);

svg.append("g")
    .attr("transform", "translate(10,410)")
    .call(xAxis);

// draw the y-axis
var yScale = d3.scaleLinear()
    .range([height,0]);
    .domain([0, 4000]);
var yAxis = d3.axisRight(yScale);

svg.append("g").attr("transform", "translate(10, 10)")
    .call(yAxis);
```

```
var width = 400;
var height = 400;

// draw the x-axis
var xScale = d3.scaleLinear()
    .domain([0, width*2])
    .range([0, width]);
var xAxis = d3.axisBottom(xScale);

svg.append("g")
    .attr("transform", "translate(10,410)")
    .call(xAxis);

// draw the y-axis
var yScale = d3.scaleLinear()
    .range([height,0]);
    .domain([0, 4000]);
var yAxis = d3.axisRight(yScale);

svg.append("g").attr("transform", "translate(10, 10)")
    .call(yAxis);
```

```
var width = 400;
var height = 400;

// draw the x-axis
var xScale = d3.scaleLinear()
    .domain([0, width*2])
    .range([0, width]);
var xAxis = d3.axisBottom(xScale);

svg.append("g")
    .attr("transform", "translate(10,410)")
    .call(xAxis);

// draw the y-axis
var yScale = d3.scaleLinear()
    .range([height,0]);
    .domain([0, 4000]);
var yAxis = d3.axisRight(yScale);

svg.append("g").attr("transform", "translate(10, 10)")
    .call(yAxis);
```

```
var width = 400;
var height = 400;

// draw the x-axis
var xScale = d3.scaleLinear()
    .domain([0, width*2])
    .range([0, width]);
var xAxis = d3.axisBottom(xScale);

svg.append("g")
    .attr("transform", "translate(10,410)")
    .call(xAxis);

// draw the y-axis
var yScale = d3.scaleLinear()
    .range([height,0]);
    .domain([0, 4000]);
var yAxis = d3.axisRight(yScale);

svg.append("g").attr("transform", "translate(10, 10)")
    .call(yAxis);
```

```
var width = 400;
var height = 400;

// draw the x-axis
var xScale = d3.scaleLinear()
    .domain([0, width*2])
    .range([0, width]);
var xAxis = d3.axisBottom(xScale);

svg.append("g")
    .attr("transform", "translate(10,410)")
    .call(xAxis);

// draw the y-axis
var yScale = d3.scaleLinear()
    .range([height,0]);
    .domain([0, 4000]);
var yAxis = d3.axisRight(yScale);

svg.append("g").attr("transform", "translate(10, 10)")
    .call(yAxis);
```

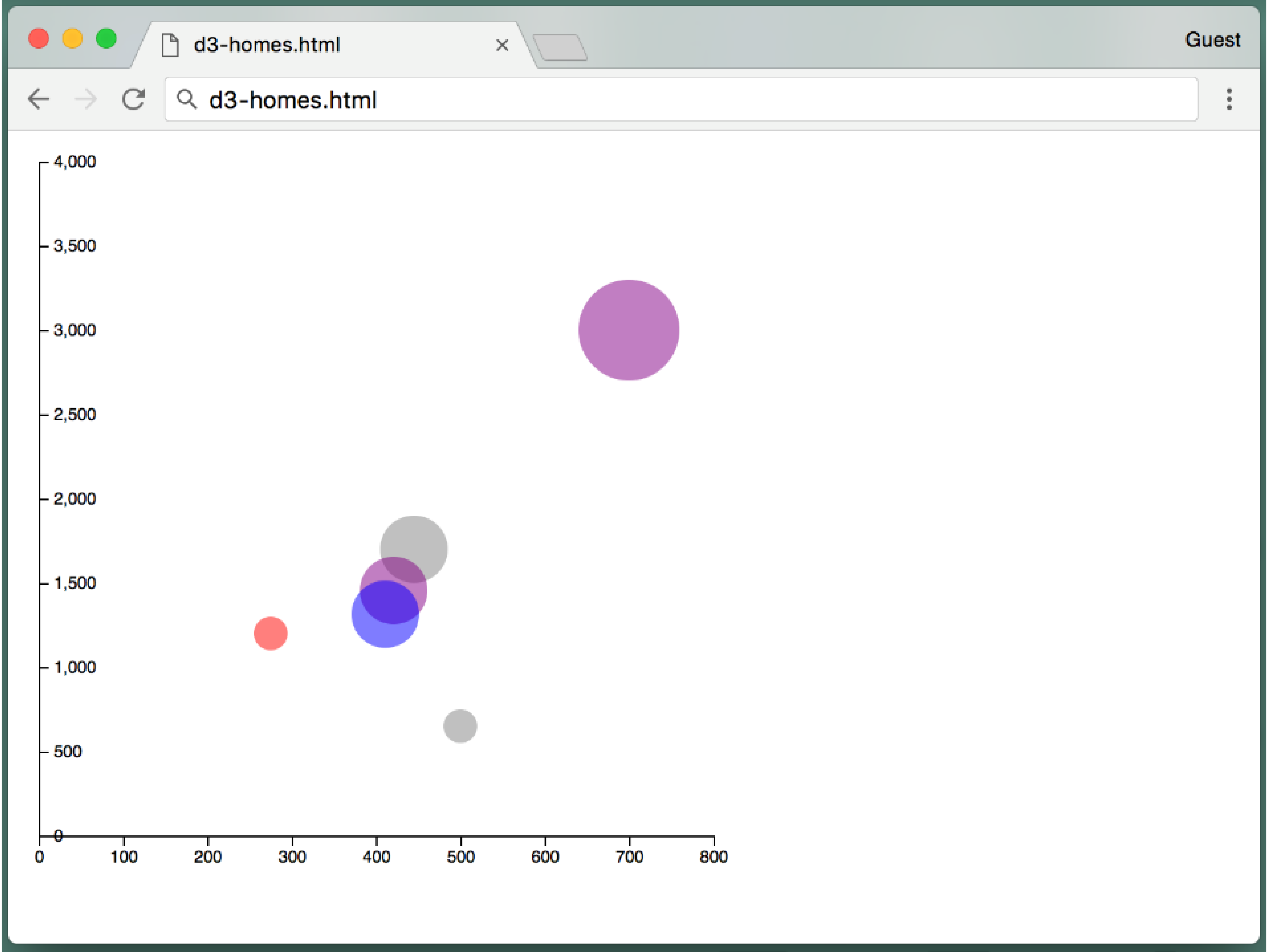


d3-homes.html
Guest
d3-homes.html
400
4,000
3,500
3,000
2,500
2,000
1,500
1,000
500
0
0
100
200
300
400
500
600
700
800

```
var width = 400;
var height = 400;

// draw the x-axis
var xScale = d3.scaleLinear()
    .domain([0, width*2])
    .range([0, width]);
var xAxis = d3.axisBottom(xScale);

svg.append("g")
    .attr("transform", "translate(10,410)")
    .call(xAxis);

// draw the y-axis
var yScale = d3.scaleLinear()
    .range([height,0]);
    .domain([0, 4000]);
var yAxis = d3.axisRight(yScale);

svg.append("g").attr("transform", "translate(10, 10)")
    .call(yAxis);
```

```
var width = 400;
var height = 400;

// draw the x-axis
var xScale = d3.scaleLinear()
    .domain([0, width*2])
    .range([0, width]);
var xAxis = d3.axisBottom(xScale);

svg.append("g")
    .attr("transform", "translate(10,410)")
    .call(xAxis);

// draw the y-axis
var yScale = d3.scaleLinear()
    .range([height,0]);
    .domain([0, 4000]);
var yAxis = d3.axisRight(yScale);

svg.append("g").attr("transform", "translate(10, 10)")
    .call(yAxis);
```

```
var width = 400;
var height = 400;

// draw the x-axis
var xScale = d3.scaleLinear()
    .domain([0, width*2])
    .range([0, width]);
var xAxis = d3.axisBottom(xScale);

svg.append("g")
    .attr("transform", "translate(10,410)")
    .call(xAxis);

// draw the y-axis
var yScale = d3.scaleLinear()
    .range([height,0]);
    .domain([0, 4000]);
var yAxis = d3.axisRight(yScale);

svg.append("g").attr("transform", "translate(10, 10)")
    .call(yAxis);
```


d3-homes.html
Guest
d3-homes.html
4,000
3,500
3,000
2,500
2,000
1,500
1,000
500
0
0
100
200
300
400
500
600
700
800

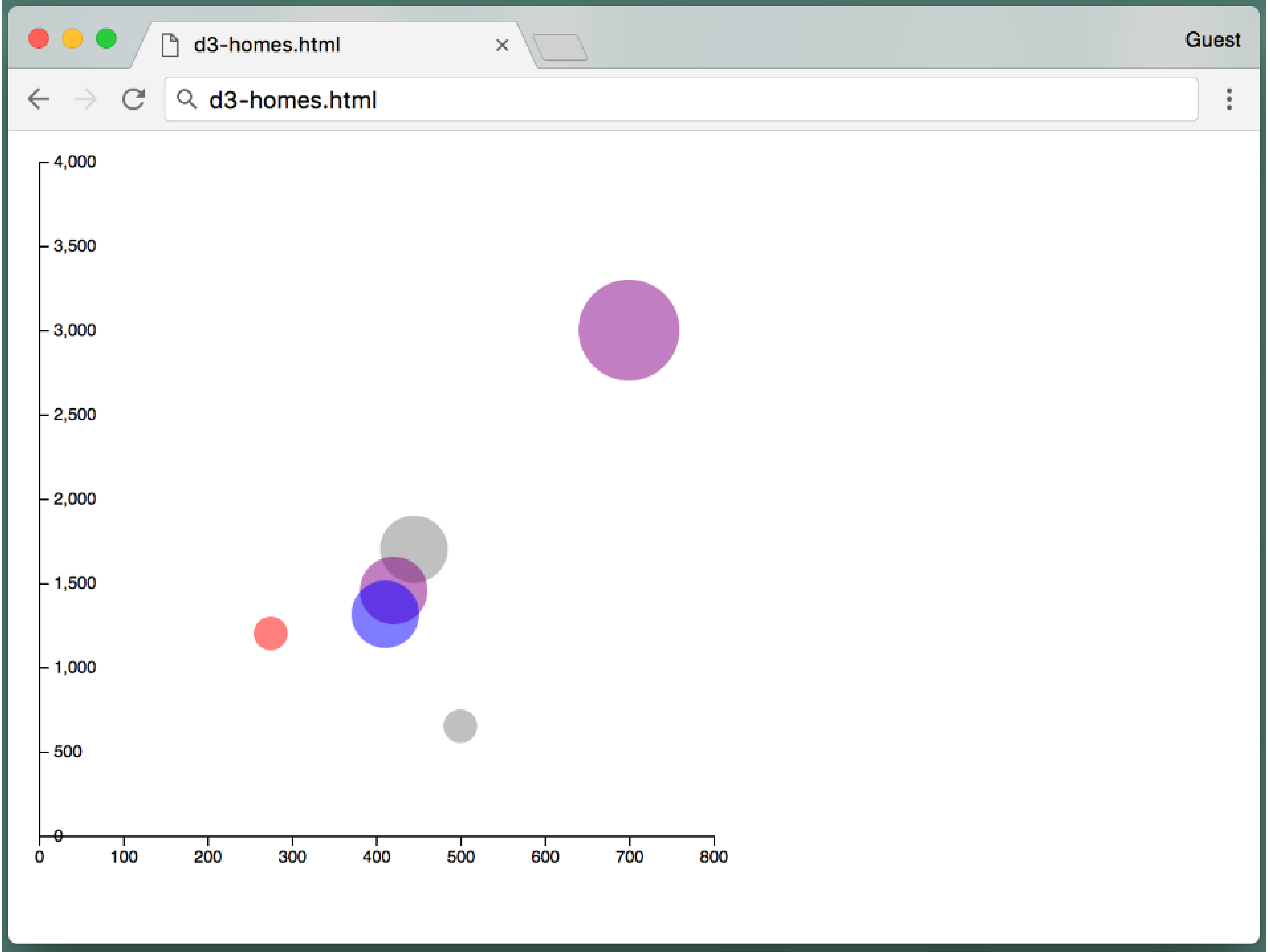
```
var width = 400;
var height = 400;

// draw the x-axis
var xScale = d3.scaleLinear()
    .domain([0, width*2])
    .range([0, width]);
var xAxis = d3.axisBottom(xScale);

svg.append("g")
    .attr("transform", "translate(10,410)")
    .call(xAxis);

// draw the y-axis
var yScale = d3.scaleLinear()
    .range([height,0]);
    .domain([0, 4000]);
var yAxis = d3.axisRight(yScale);

svg.append("g").attr("transform", "translate(10, 10)")
    .call(yAxis);
```

```
var width = 400;
var height = 400;

// draw the x-axis
var xScale = d3.scaleLinear()
    .domain([0, width*2])
    .range([0, width]);
var xAxis = d3.axisBottom(xScale);

svg.append("g")
    .attr("transform", "translate(10,410)")
    .call(xAxis);

// draw the y-axis
var yScale = d3.scaleLinear()
    .range([height,0]);
    .domain([0, 4000]);
var yAxis = d3.axisRight(yScale);

svg.append("g").attr("transform", "translate(10, 10)")
    .call(yAxis);
```

```
var width = 400;
var height = 400;

// draw the x-axis
var xScale = d3.scaleLinear()
    .domain([0, width*2])
    .range([0, width]);
var xAxis = d3.axisBottom(xScale);

svg.append("g")
    .attr("transform", "translate(10,410)")
    .call(xAxis);

// draw the y-axis
var yScale = d3.scaleLinear()
    .range([height,0]);
    .domain([0, 4000]);
var yAxis = d3.axisRight(yScale);

svg.append("g").attr("transform", "translate(10, 10)")
    .call(yAxis);
```

```
var width = 400;
var height = 400;

// draw the x-axis
var xScale = d3.scaleLinear()
    .domain([0, width*2])
    .range([0, width]);
var xAxis = d3.axisBottom(xScale);

svg.append("g")
    .attr("transform", "translate(10,410)")
    .call(xAxis);

// draw the y-axis
var yScale = d3.scaleLinear()
    .range([height,0]);
    .domain([0, 4000]);
var yAxis = d3.axisRight(yScale);

svg.append("g").attr("transform", "translate(10, 10)")
    .call(yAxis);
```

```
var width = 400;
var height = 400;

// draw the x-axis
var xScale = d3.scaleLinear()
    .domain([0, width*2])
    .range([0, width]);
var xAxis = d3.axisBottom(xScale);

svg.append("g")
    .attr("transform", "translate(10,410)")
    .call(xAxis);

// draw the y-axis
var yScale = d3.scaleLinear()
    .range([height,0]);
    .domain([0, 4000]);
var yAxis = d3.axisRight(yScale);

svg.append("g").attr("transform", "translate(10, 10)")
    .call(yAxis);
```


d3-homes.html
Guest
d3-homes.html
4,000
3,500
3,000
2,500
2,000
1,500
1,000
500
0
0
100
200
300
400
500
600
700
800

Accessing Data with D3.js

- D3.js can easily access data on the Web that is available through RESTful APIs

```
var values = [];

var URL = . . .

d3.json(URL, (response) => {
   // populate the values from the data in
   // response that comes back from request
   . . .
});

// now use values with D3 functions

var svg = d3.select("svg");
var selection = svg.selectAll("g")
    .data(values)
    . . .
```

Accessing Data with D3.js

- D3.js can easily access data on the Web that is available through RESTful APIs

```
var values = [];

var URL = . . .

d3.json(URL, (response) => {
   // populate the values from the data in
   // response that comes back from request
   . . .
});

// now use values with D3 functions

var svg = d3.select("svg");
var selection = svg.selectAll("g")
    .data(values)
    . . .
```

Accessing Data with D3.js

- D3.js can easily access data on the Web that is available through RESTful APIs

```
var values = [];

var URL = . . .

d3.json(URL, (response) => {
   // populate the values from the data in
   // response that comes back from request
   . . .
});

// now use values with D3 functions

var svg = d3.select("svg");
var selection = svg.selectAll("g")
    .data(values)
    . . .
```

Accessing Data with D3.js

- D3.js can easily access data on the Web that is available through RESTful APIs

```
var values = [];

var URL = . . .

d3.json(URL, (response) => {
   // populate the values from the data in
   // response that comes back from request
   . . .
});

// now use values with D3 functions

var svg = d3.select("svg");
var selection = svg.selectAll("g")
    .data(values)
    . . .
```

Accessing Data with D3.js

- D3.js can easily access data on the Web that is available through RESTful APIs

```
var values = [];

var URL = . . .

d3.json(URL, (response) => {
   // populate the values from the data in
   // response that comes back from request
   . . .
});

// now use values with D3 functions

var svg = d3.select("svg");
var selection = svg.selectAll("g")
    .data(values)
    . . .
```

Accessing Data with D3.js

- D3.js can easily access data on the Web that is available through RESTful APIs

```
var values = [];

var URL = . . .

d3.json(URL, (response) => {
    // populate the values from the data in
    // response that comes back from request
    . . .
});

// now use values with D3 functions

var svg = d3.select("svg");
var selection = svg.selectAll("g")
     .data(values)
     . . .
```

Accessing Data with D3.js

- D3.js can easily access data on the Web that is available through RESTful APIs

```
var values = [];

var URL = . . .

d3.json(URL, (response) => {
    // populate the values from the data in
    // response that comes back from request
    . . .
});

// now use values with D3 functions

var svg = d3.select("svg");
var selection = svg.selectAll("g")
     .data(values)
     . . .
```

Accessing Data with D3.js

- D3.js can easily access data on the Web that is available through RESTful APIs

```
var values = [];

var URL = . . .

d3.json(URL, (response) => {
   // populate the values from the data in
   // response that comes back from request
   . . .
});

// now use values with D3 functions

var svg = d3.select("svg");
var selection = svg.selectAll("g")
    .data(values)
    . . .
```

Summary

- D3.js allows us to generate HTML and SVG elements based on data
- We can apply functions to data sets to generate graphical elements, e.g. charts
- The data used by D3.js can include objects
- You can easily access data online using D3.js functions

Review: Week 3

- **React**
  - library and framework for creating reusable, modular components
  - can render themselves based on their state
  - can be combined and work together

- **D3.js**
  - library for generating HTML and SVG based on data

- **ES6**: more recent version of JavaScript