In [1]:

```
var data1;

async function loadData() {
  const data = await
d3.csv('https://raw.githubusercontent.com/ywchiu/riii/master/data/house-prices.csv');
  const Price=[] ;
  const SqFt =[] ;
  const Bedrooms=[];
  const Bathrooms =[];
  const Offers =[];
  data.forEach(d => {
    SqFt.push(parseFloat(d.SqFt));
    Price.push(parseFloat(d.Price));
    Bedrooms.push(parseInt(d.Bedrooms));
    Bathrooms.push(parseInt(d.Bathrooms));
    Offers.push(parseInt(d.Offers));
  });
  return { SqFt, Price,Bedrooms,Bathrooms,Offers };
}

async function run() {
  data1 = await loadData();
  return data;
}

// Wrap the run() function in a Promise and wait for it to finish before trying to access
plotData and layout:
new Promise(resolve => {
  run().then(() => {
    show('data loaded');
    resolve();
  });
}).then(() => {
show('success');

});
```

In [2]:

```
var tableData = [{  type: 'table',  header: {    values: [['<b>SqFit</b>'],
['<b>Price</b>'],['<b>Bedrooms</b>'], ['<b>Bathrooms</b>'],['<b>Offers</b>']],
    align: ['center', 'center'],
    line: {width: 1, color: 'black'},
    fill: {color: '#506784'},
    font: {family: 'Arial', size: 12, color: 'white'} ,
  },
  cells: {
    values: [data1.SqFt, data1.Price, data1.Bedrooms,data1.Bathrooms,data1.Offers],
    align: ['center', 'center'],
    line: {color: '#506784', width: 1},
    font: {family: 'Arial', size: 11, color: ['#506784']}
  }
}];
var layoutx = {
```

```
 height: 600,
}
```

In [3]:

```
show_graph(tableData,layoutx);
```

Out[3]:

x

SqFit	Price	Bedrooms	Bathrooms	Offers
1790	114300	2	2	2
2030	114200	4	2	3
1740	114800	3	2	1
1980	94700	3	2	3
2130	119800	3	3	3
1780	114600	3	2	2
1830	151600	3	3	3
2160	150700	4	2	2
2110	119200	4	2	3
1730	104000	3	3	3
2030	132500	3	2	3
1870	123000	2	2	2
1910	102600	3	2	4
2150	126300	3	3	5
2590	176800	4	3	4
1780	145800	4	2	1
2190	147100	3	3	4
1990	83600	3	3	4
1700	111400	2	2	1

In [4]:

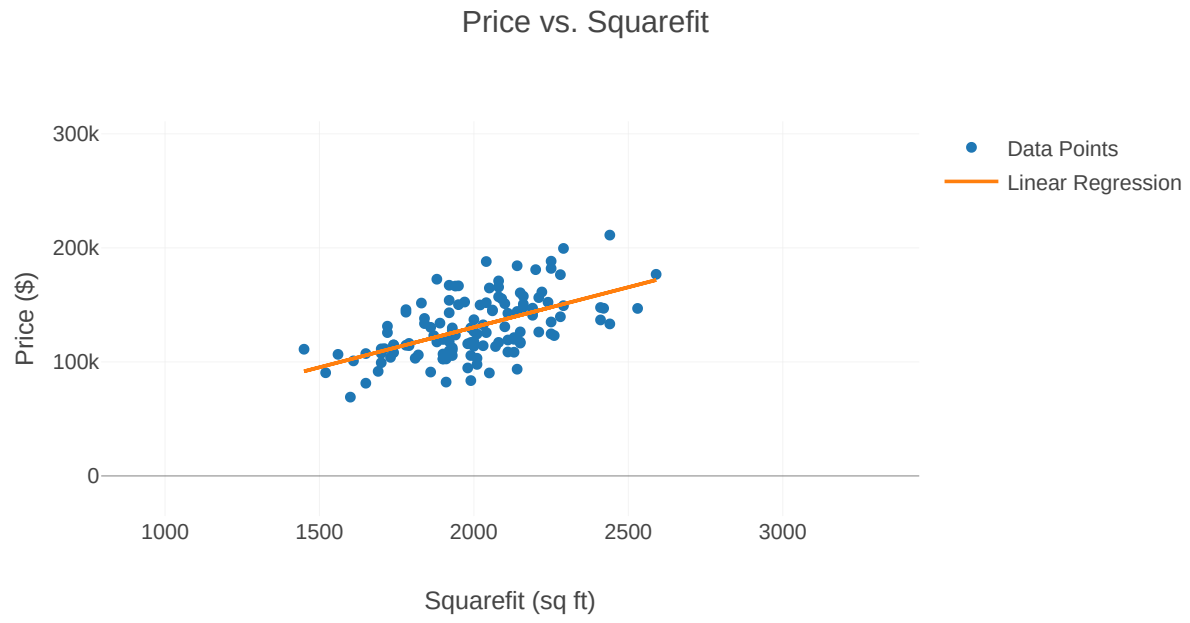
```
// Calculate the regression line using linear regression formula
const x= data1.SqFt;
const y= data1.Price;
const n = x.length;
const sum_x = x.reduce((a, b) => a + b, 0);
const sum_y = y.reduce((a, b) => a + b, 0);
const sum_xy = x.map((xi, i) => xi * y[i]).reduce((a, b) => a + b, 0);
const sum_xx = x.map(xi => xi * xi).reduce((a, b) => a + b, 0);
const slope = (n * sum_xy - sum_x * sum_y) / (n * sum_xx - sum_x * sum_x);
const intercept = (sum_y - slope * sum_x) / n;

// Create the trace for the data points
const dataTrace = {
  x: x,
  y: y,
  mode: 'markers',
  name: 'Data Points'
};

// Create the trace for the regression line
const lineTrace = {
  x: x,
  y: x.map(xi => slope * xi + intercept),
```

```
  mode: 'lines',
  name: 'Linear Regression'
};

// Create the layout
var layout2 = {
          title: 'Price vs. Squarefit',
          yaxis: {
            title: 'Price ($)'
          }
        };

// Combine the traces into an array
var data2 = [dataTrace, lineTrace];
```

Linear Regression JS

In [5]:
```
show_graph(data2,layout2)
```

Out[5]:



In [6]:
```

```