

CSE 564: MINI PROJECT I

PROJECT REPORT

SHREYA SAMJNAN SINGH, ID: 112675992

- Youtube link: <https://youtu.be/bDL9HwORii0>
- The Dataset chosen is AMES HOUSING DATASET which is available on Kaggle. It has 1460 rows. I have chosen only 16 attributes out of more than 30.
- Data Preparation:
 - CSV stores all rows as objects. Therefore, to deal with this, mapped each individual column when it was selected to an array of its values.
 - Prepared data depending on whether it was categorical or numerical.
- Data preparation for Categorical Variables:
 - Since categorical data consists of text, created a dictionary of a column with each of its category as keys and its total count as values.

```
//loading data
d3.csv("data/house.csv", function(error, data) {
  //Data Preparation
  //mapping all the row values of a particular column
  var map = data.map(function(d) { return d.MSZoning; })
  // creating dictionary for all distinct categories
  var counts = {};
  for (var i = 0; i < map.length; i++) {
    // summing up all values in a particular category
    counts[map[i]] = 1 + (counts[map[i]] || 0);
  }
}
```

- Then pushed all the key and value pairs as two separate dictionaries into an array to create an array of dictionaries. As follows:

```
var a = []
// creating an array of dictionaries, with all categories & their respective frequency
for (key in counts){
  a.push({"categories": key, "values": counts[key]})
}
```

- Data Preparation for Numerical Variables:
 - Since the numerical variables may be mapped in a random order, applied sorting function.

```
//loading data
d3.csv("data/house.csv", type, function(error, data) {
  //Data Preparation
  //mapping all the row values of a particular column
  var map = data.map(function(d) { return d.YearBuilt; })
  // Arranging numerical values in an ascending order
  map.sort(function(a, b){return a-b});
}
```

- Using the built in layout function for histogram to generate binned data.

```
bin = 10 // setting default bin value as 10 which we will update later

// Using built in histogram function and passing our data to it
var data = d3.layout.histogram()
  .bins(x.ticks(bin))
  (map);
```

- Change of bin width (increase/decrease):

- I'm using an html slider for this particular functionality in my program.

```
<!--Creating a slider to change bin width-->
<div class="box">
  <center><div id="value"></div></center>
</div>
<div class="container">
  <input type="range" min="0" max="70" value="10" class="slider" id="slider">
</div>
```

- Using javascript to read the value of slider & updating my histogram accordingly.

```
// Listen to the slider
var val = document.getElementById('value'); // storing value of the slider
d3.select("#slider").on("change", function(d) { // notices change in slider
  d3.select('.box').selectAll("text").remove() // removes previously selected value
  selectedValue = this.value
  d3.select(".box").append("text").attr('class', 'try').text(function () {
    return selectedValue; // returns new value
  })
  updateChart(selectedValue) // returns value to function to update histogram
})
```

- Mouse Hover & Mouse Out for Categorical Data:

- For Bar Chart there is an increase in width and height of the bar, bar also changes color when selected and shows the frequency at top of the bar.
- On Mouse out, the bar returns to its default color and size. The tooltip showing frequency is removed as well.

```
function onMouseOver(d){
  d3.select(this).style("fill", "#cc3f0c")
  .attr('width', x.rangeBand() + 5) //increasing width of the bar
  .attr("y", function(d) { return y(d.values) - 10; })
  .attr("height", function(d) { return height - y(d.values) + 10; }) // increasing height
  tip.html("<span>" + d.values + "</span>") // display frequency of bar
  tip.show()
}

function onMouseOut(d){
  // Returning bar values to default
  d3.select(this).style("fill", function(d) { return colorScale(d.values) });
  d3.select(this)
    .attr('width', x.rangeBand())
    .attr("height", function(d) { return height - y(d.values); })
    .attr("y", function(d) { return y(d.values); })
    tip.hide() // hide frequency of bar
}
```

- Mouse Hover & Mouse Out for Numerical Data:

- There is focus on the selected histogram bar with color change and tool tip showing frequency. There is no height and width change for the histogram as histograms are not reordered.
- On Mouse out, the bar returns to its default color and size. The tooltip showing frequency is removed as well.

```
function onMouseOver(d){
  d3.select(this).select("rect").style("fill", "#cc3f0c")
  tip.html("<span>" + d.y + "</span>")
  tip.show()
}

function onMouseOut(d){
  d3.select(this).select("rect").style("fill", function(d) { return colorScale(d.y) })
  tip.hide()
}
```

- Added Functionalities:

- I'm using color scale to calculate color for the bars. This gives a dark color for bars with max higher frequency count and a light color for bar with smallest frequency count. This has been applied to both categorical & numerical data.

```
// creating a colorscale effect, darker for max value and lighter for min value
var colorScale = d3.scale.linear()
    .domain([0,d3.max(a, function(d) { return d.values })])
    .range([d3.rgb(color).brighter(), d3.rgb(color).darker()]);
```

- Tool tip has been added for clear representation of frequency count on top of the bar. A script is being accessed. Along with the following code for both numerical & categorical data.

```
// creating tip variable and calling it in svg
var tip = d3.tip()
    .attr('class', 'd3-tip')
    .offset([-5, 0])
    svg.call(tip)
```

- Drop Down Selection:

- I have made a control selection drop down with options and connected that with javascript so that different variables can be selected.

```
<div><h1 style="text-align: left; color: black"> AMES HOUSING DATASET</h1></div>
<div class="form-inline custom-centered">
    <label for="controlSelection">Select Variable: &nbsp;</label>
    <select class="form-control" id="controlSelection">
        <option value="">Select a value</option>
        <option value="MSZoning">MSZoning</option>
        <option value="YearBuilt">Year Built</option>
        <option value="SaleCondition">Sale Condition</option>
        <option value="SalePrice">Sale Price</option>
        <option value="SaleType">Sale Type</option>
        <option value="GarageArea">Garage Area</option>
        <option value="Electrical">Electrical</option>
        <option value="firstFlrSF">First floor surface area</option>
        <option value="Foundation">Foundation of the House</option>
        <option value="GarageType">GarageType</option>
        <option value="GrLivArea">Ground Living Area</option>
        <option value="OverallCond">Overall Condition of House</option>
        <option value="OverallQual">Overall Quality of House</option>
        <option value="RoofStyle">Roof Style</option>
        <option value="BsmtQual">Basement Quality</option>
        <option value="TotRmsAbvGrd">Total Rooms Above Ground</option>
    </select>
</div>
```

- As soon as the control text is selected, the value is passed to javascript function change which notices this and passes this variable to column. Along with that a categorical data array is made which helps in deciding if selected value is categorical or numerical and plots the chart accordingly.

```

//data preparation
// Selecting button will record change in selection of variable
d3.select("#controlSelection").on("change", function() {
  d3.selectAll('svg').remove(); // remove any old charts
  column = this.value; // selecting a particular column
  map = [];
  // Creating array consisting of all categorical data
  categorical_data = ["MSZoning", "SaleCondition", "SaleType", "Electrical", "Foundation",
    "Neighborhood", "GarageType", "RoofStyle", "BsmtQual"];
  if (categorical_data.indexOf(column) != -1) {
    // if categorical, then map the values of the column and call function drawBarChart
    for (let index = 0; index < data.length; index++) {
      map.push(data[index][column]);
    }
    drawBarChart(column, map);
  } else {
    // otherwise map values of column and call drawHistogram
    for (let index = 0; index < data.length; index++) {
      map.push(parseInt(data[index][column]));
    }
    drawHistogram(column, map);
  }
});

```

- References:

- <http://bl.ocks.org/nnattawat/8916402>
- <https://observablehq.com/@d3/bar-chart>
- <https://alignedleft.com/tutorials/d3/making-a-bar-chart>
- <http://bl.ocks.org/caged/6476579>
- <http://labratrevenge.com/d3-tip/javascripts/d3.tip.v0.6.3.js>
- https://www.d3-graph-gallery.com/graph/density_slider.html
- https://www.w3schools.com/html/html_form_input_types.asp
- <https://coolors.co/d8cbc7-2a324b-fff6ed-cc3f0c-9a6d38>