CSE 564 Project 1 Report

YouTube Link - https://youtu.be/4IwRuL1BIPc

1. Dataset description

- The data set used is called the “Ames Housing Dataset”. It describes the sale of individual residential property in Ames, Iowa from 2006 to 2010. The data set contains 2930 observations and a large number of explanatory variables (23 nominal, 23 ordinal, 14 discrete, and 20 continuous) involved in assessing home values.
- For this project, I have chosen 15 variables - SalePrice, GrLivArea, OverallQual, SaleCondition, BldgType, YearBuilt, GarageCars, LotArea, HeatingQC, KitchenQual, MiscVal, YearRemodAdd, YrSold, RoofStyle, SaleType.
- The Pandas library in Python was used to filter out/drop any “NULL” values in the dataset which resulted in a reduction of observations to 1460 rows**.**
- The final filtered data frame was again stored as a new CSV to be used in the project.
- The other datasets for housing prices like the Boston Housing Dataset have very fewer observations as compared to this dataset which makes it a very good candidate for finding trends in housing prices.

2. Functionalities implemented

2.1. Data Loading

- Data loading was performed using the **CSV** module of D3 v4. The whole CSV is read into memory as soon as the webpage is loaded for the first time.
- The following code snippet was used for the same -

```
d3.csv("data/shortened.csv", function(error, data) {
 if (error) {
             throw error;
       }
// remaining logic
});
```

2.2. Menu for variables

- Bootstrap’s various classes have been used to increase the aesthetics of the webpage.
- The following snippet was used here -

HTML -

```
<div class="col-md-2">
     <select class="form-control", id="dropdown">
     </select>
</div>
```

Javascript -

```
column_names = Object.keys(data[0]);
column_names.unshift("Choose variable");

d3.select("#dropdown")
    .selectAll("option")
    .data(column_names)
    .enter()
    .append("option")
    .text(function (d) {
        return d;
    });
```

2.3. Showing description based on selected dropdown column

- If the value “Choose variable” is selected, a short description of the dataset is displayed.
- If a numeric column/variable is selected, a short description of the variable is displayed.
- If a categorical column/variable is selected, a short description of the variable, as well as the categories, is displayed.
- The bins slider is also shown only when a numeric variable is selected.
- The following snippet is used for the same -

```
colName = d3.select(this).property("value");

if (colName == "Choose variable") {

    // show dataset description
    d3.select("#dataset-desc")
        .classed("d-none", false);

    // remove categorical description if any
    d3.select("#categorical_description")
        .classed("d-none", true)
        .select("p")
        .selectAll("p")
        .remove();

    // remove numerical description if any
    d3.select("#numerical_description")
```

```
            .classed("d-none", true)
            .select("p")
            .selectAll("p")
            .remove();

        // hide bin slider
        hide_text_box(true);
    }

    else if (numerical_cols.has(colName)) {
        console.log("Numerical found");

        // remove dataset description
        d3.select("#dataset-desc")
            .classed("d-none", true);

        // remove categorical description if any

        // show bin slider
        hide_text_box(false);

        generate_numerical_graph(colName);
    }
    else {
        console.log("Categorical found");

        // remove dataset description

        // remove numerical description if any

        // hide bin slider
        hide_text_box(true);

        generate_categorical_graph(colName);
    }
```

2.4. Bar chart and histogram

- The following snippet was used for adding the rectangle bars in both, bar chart as well as the histogram -

```
g.append("g")
.selectAll(".bar")
.data(graph_input)
.enter().append("rect")
.attr("class", "bar")
.on("mouseover", onMouseOver) // Add listener for mouseover event
.on("mouseout", onMouseOut)  // Add listener for mouseout event
.attr("x", function(d) { return x(d.key);})
.attr("y", function(d) { return y(d.freq);})
```

```
    .attr("width", x.bandwidth())
    .transition()
    .ease(d3.easeLinear)
    .duration(400)
    .delay(function (d, i) { return i * 50;})
    .attr("height", function(d) { return height - y(d.freq);});
```

- The “graph_input” for the histogram is generated by the following snippet which makes use of d3.histogram -

```
var histogram = d3.histogram()
    .value(function(d) { return d.value; })
    .domain(x.domain())
    .thresholds(X.ticks(nBin));

// And apply this function to data to get the bins
var bins = histogram(value_list);
```

  “bins” has the details about bin-ranges as well as the number of observations/records in that range. Hence we just change the snippet for adding rectangles a bit for the histograms.

2.5. On MouseOver functionality

- The following code was used for increasing the height and width as well as to display the current value on top of the bar -

```
function onMouseOver(d, i) {
    d3.select(this).attr('class', 'highlight');
    d3.select(this)
    .transition()     // adds animation
    .duration(400)
    .attr('width', x.bandwidth() + 5)
    .attr("y", function(d) { return y(d.freq) - 10; })
    .attr("height", function(d) { return height - y(d.freq)+ 10; });
    g.append("text")
    .attr('class', 'val')
    .attr('x', function() { return x(d.key);})
    .attr('y', function() { return y(d.freq) - 15;})
    .text(function() {return [d.freq];});
}
```

2.6. On MouseOut functionality

- The following code was used for bringing the size of the bar back to normal -

```
function onMouseOut(d, i) {
    d3.select(this).attr('class', 'bar');
    d3.select(this)
    .transition()     // adds animation
    .duration(400)
    .attr('width', x.bandwidth())
    .attr("y", function(d) { return y(d.freq); })
    .attr("height", function(d) { return height - y(d.freq); });
    d3.selectAll('.val')
    .remove()
}
```

2.7. Bin slider to change the number of bins

- To change the number of bins, as clicking and moving the mouse pointer anywhere on the screen would be inconvenient in some cases especially if the user is on a mobile device.
- Moreover, a slider gives more freedom to increase or decrease the bins over a long-range. Doing so by using a mouse pointer would be slow and complicated.
- The slider provides more visual feedback. On changing the value of the slider, the JavaScript function called “update” is used to update the graph.
- The following snippets were used here -

  HTML -

```
<div class="col-md-5">
   <input  type="range"  class="custom-range"  min="0"  max="50"
id="slider-value" value="10">
 </div>
```

  JavaScript -

```
d3.select("#slider-value").on("input", function() {
    d3.select("h5").text(this.value);
    update(+this.value);
});
```

References:

1. De Cock, Dean. "Ames, Iowa: Alternative to the Boston housing data as an end of semester regression project." Journal of Statistics Education 19.3 (2011).