CSCE 5320 Scientific Data Visualization
ICE-10
Interaction Techniques

I selected the Car Sales Dataset for this assignment and inserted the data I had into the GitHub Gist. I will now access my data using Jupyter Notebook by using the raw data path.

I retrieved the data and imported it into a pandas data frame using the read_csv command in the pandas library.

GitHub Gist:
https://gist.githubusercontent.com/nehabaddam/1f47243bf7cd359b25e88d9c100b8248/raw/cc814c12211b47b99233144a642dd8ada0fb52a5/car_sales.csv

This data set contains details about various automotive sales. I got this dataset from Kaggle, and it is basically sales data for cars with plenty of car-related variables. It fundamentally contains records of all car models, sales data, price data, and feature data from various automotive manufacturing companies. There are about 157 records in it. This information can be used to show the relationship between car sales and various factors, such as horsepower, price, and car features.

```
Manufacturer,Model,Sales_in_thousands,__year_resale_value,Vehicle_type,Price_in_thousands,Engine_size,Horsepower,Wheelbase,Width,Length,Curb_weight,Fuel_capacity,Fuel_efficiency,Latest_Launch,Power_perf_factor
Acura,Integra,16.919,16.36,Passenger,21.5,1.8,140,101.2,67.3,172.4,2.639,13.2,28,2/2/2012,58.28014952
Acura,TL,39.384,19.875,Passenger,28.4,3.2,225,108.1,70.3,192.9,3.517,17.2,25,6/3/2011,91.37077766
Acura,CL,14.114,18.225,Passenger,,3.2,225,106.9,70.6,192,3.47,17.2,26,1/4/2012,
Acura,RL,8.588,29.725,Passenger,42,3.5,210,114.6,71.4,196.6,3.85,18,22,3/10/2011,91.38977933
Audi,A4,20.397,22.255,Passenger,23.99,1.8,150,102.6,68.2,178,2.998,16.4,27,10/8/2011,62.7776392
Audi,A6,18.78,23.555,Passenger,33.95,2.8,200,108.7,76.1,192,3.561,18.5,22,8/9/2011,84.56510502
Audi,A8,1.38,39,Passenger,62,4.2,310,113,74,198.2,3.902,23.7,21,2/27/2012,134.6568582
BMW,323i,19.747,,Passenger,26.99,2.5,170,107.3,68.4,176,3.179,16.6,26,6/28/2011,71.19120671
BMW,328i,9.231,28.675,Passenger,33.4,2.8,193,107.3,68.5,176,3.197,16.6,24,1/29/2012,81.87706856
BMW,528i,17.527,36.125,Passenger,38.9,2.8,193,111.4,70.9,188,3.472,18.5,25,4/4/2011,83.9987238
Buick,Century,91.561,12.475,Passenger,21.975,3.1,175,109,72.7,194.6,3.368,17.5,25,11/2/2011,71.18145132
Buick,Regal,39.35,13.74,Passenger,25.3,3.8,240,109,72.7,196.2,3.543,17.5,23,9/3/2011,95.63670253
Buick,Park Avenue,27.851,20.19,Passenger,31.965,3.8,205,113.8,74.7,206.8,3.778,18.5,24,3/23/2012,85.82840825
Buick,LeSabre,83.257,13.36,Passenger,27.885,3.8,205,112.2,73.5,200,3.591,17.5,25,7/23/2011,84.25452581
Cadillac,DeVille,63.729,22.525,Passenger,39.895,4.6,275,115.3,74.5,207.2,3.978,18.5,22,2/23/2012,113.8545976
Cadillac,Seville,15.943,27.1,Passenger,44.475,4.6,275,112.2,75,201,,18.5,22,4/29/2011,115.6213578
Cadillac,Eldorado,6.536,25.725,Passenger,39.665,4.6,275,108,75.5,200.6,3.843,19,22,11/27/2011,113.7658739
Cadillac,Catera,11.185,18.225,Passenger,31.01,3,200,107.4,70.3,194.8,3.77,18,22,9/28/2011,83.48309358
Cadillac,Escalade,14.785,,Car,46.225,5.7,255,117.5,77,201.2,5.572,30,15,4/17/2012,109.5091165
Chevrolet,Cavalier,145.519,9.25,Passenger,13.26,2.2,115,104.1,67.9,180.9,2.676,14.3,27,8/17/2011,46.36334747
Chevrolet,Malibu,135.126,11.225,Passenger,16.535,3.1,170,107,69.4,190.4,3.051,15,25,3/19/2012,67.31446216
Chevrolet,Lumina,24.629,10.31,Passenger,18.89,3.1,175,107.5,72.5,200.9,3.33,16.6,25,5/24/2011,69.9913956
Chevrolet,Monte Carlo,42.593,11.525,Passenger,19.39,3.4,180,110.5,72.7,197.9,3.34,17,27,12/22/2011,72.03091719
Chevrolet,Camaro,26.402,13.025,Passenger,24.34,3.8,200,101.1,74.1,193.2,3.5,16.8,25,10/23/2011,81.11854333
Chevrolet,Corvette,17.947,36.225,Passenger,45.705,5.7,345,104.5,73.6,179.7,3.21,19.1,22,5/12/2012,141.14115
Chevrolet,Prizm,32.299,9.125,Passenger,13.96,1.8,120,97.1,66.7,174.3,2.398,13.2,33,9/11/2011,48.2976361
Chevrolet,Metro,21.855,5.16,Passenger,9.235,1,55,93.1,62.6,149.4,1.895,10.3,45,4/13/2012,23.27627233
Chevrolet,Impala,107.995,,Passenger,18.89,3.4,180,110.5,73,200,3.389,17,27,6/18/2011,71.83803944
Chrysler,Sebring Coupe,7.854,12.36,Passenger,19.84,2.5,163,103.7,69.7,190.9,2.967,15.9,24,1/16/2012,65.95718396
Chrysler,Sebring Conv.,32.775,14.18,Passenger,24.495,2.5,168,106,69.2,193,3.332,16,24,11/17/2011,69.52135505
Chrysler,Concorde,31.148,13.725,Passenger,22.245,2.7,200,113,74.4,209.1,3.452,17,26,6/6/2012,80.02378204
Chrysler,Cirrus,32.306,12.64,Passenger,16.48,2,132,108,71,186,2.911,16,27,10/6/2011,53.56619987
Chrysler,LHS,13.462,17.325,Passenger,28.34,3.5,253,113,74.4,207.7,3.564,17,23,5/8/2012,101.3292807
Chrysler,Town & Country,53.48,19.54,Car,,,,,,,,,7/13/2011,
Chrysler,300M,30.696,,Passenger,29.185,3.5,253,113,74.4,197.8,3.567,17,23,2/10/2012,101.6552441
Dodge,Neon,76.034,7.75,Passenger,12.64,2,132,105,74.4,174.4,2.567,12.5,29,12/12/2011,52.08489875
Dodge,Avenger,4.734,12.545,Passenger,19.045,2.5,163,103.7,69.1,190.2,2.879,15.9,24,7/1/2012,65.65050834
Dodge,Stratus,71.186,10.185,Passenger,20.23,2.5,168,108,71,186,3.058,16,24,10/31/2011,67.87610784
Dodge,Intrepid,88.028,12.275,Passenger,22.505,2.7,202,113,74.7,203.7,3.489,17,,6/2/2012,80.83147017
Dodge,Viper,0.916,58.47,Passenger,69.725,8,450,96.2,75.7,176.7,3.375,19,16,8/7/2011,188.144323
Dodge,Ram Pickup,227.061,15.06,Car,19.46,5.2,230,138.7,79.3,224.2,4.47,26,17,3/6/2012,90.21170005
Dodge,Ram Wagon,16.767,15.51,Car,21.315,3.9,175,109.6,78.8,192.6,4.245,32,15,1/6/2012,71.13529161
Dodge,Ram Van,31.038,13.425,Car,18.575,3.9,175,127.2,78.8,208.5,4.298,32,16,7/26/2012,70.07832154
Dodge,Dakota,111.313,11.26,Car,16.98,2.5,120,131,71.5,215,3.557,22,19,11/25/2011,49.64500177
Dodge,Durango,101.323,,Car,26.31,5.2,230,115.7,71.7,193.5,4.394,25,17,6/27/2012,92.85412522
```

I. MAKING INTERACTIVE VISUALIZATIONS

1.1 Please show the standard scatter plot to which you are going to add interaction, submit the screenshot of the graph, and describe your data/graph including all labels and legends.

I am plotting a scatter plot using my Car Sales dataset. To visualize my data and create an interactive chart, using the Altair library. I am using only the Manufacturer, Sales_in_thousands, and Price_in_thousands columns to show the relationship between car price and sales, grouped by manufacturer.

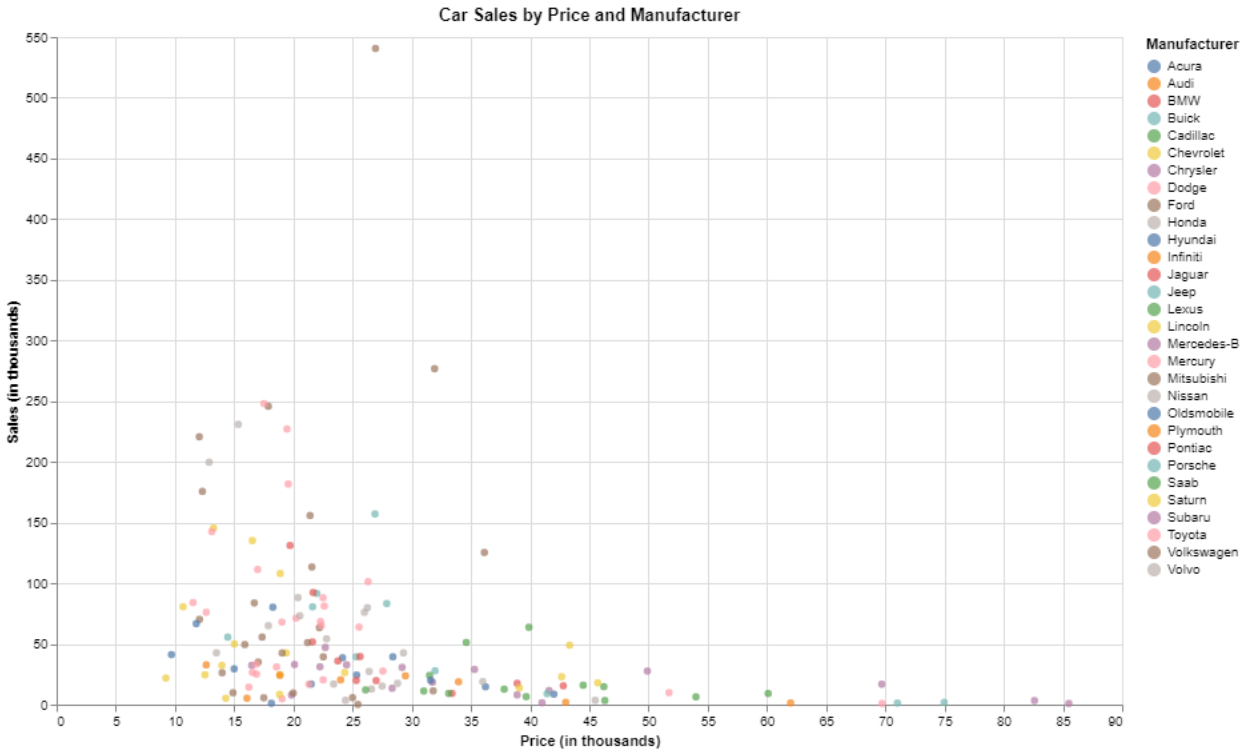
Now to the scatter chart, we have encoded the x and y variables that are mapped to the Price_in_thousands and Sales_in_thousands columns of the car sales dataset, respectively. We are using color to encode the Manufacturer column, which displays data points with respect to each manufacturer.

We use properties to indicate the size of the chart and we even display the title.

Finally, we are using the scatter to display the chart.

```
# Create the scatter plot
scatter = alt.Chart(carsales).mark_circle().encode(
    x=alt.X('Price_in_thousands:Q', title='Price (in thousands)'),
    y=alt.Y('Sales_in_thousands:Q', title='Sales (in thousands)'),
    color=alt.Color('Manufacturer', title='Manufacturer'),
).properties(
    width=800,
    height=500,
    title='Car Sales by Price and Manufacturer'
)
# Display the scatter plot
scatter
```

The chart is shown below. We can see all the data points showing the relation between the sales and price of all manufacturers. We can see that the major datapoints are located between the 10 to 45-thousand-dollar Price and between the 0-150 thousand sales. We can see that the sales decreased with the increase in the price with respect to all manufacturers. Each manufacturer is displayed using a different color.



1.2 Which library/ package are you going to use for interactive visualization in this lab? Simply describe them (such as Matplotlib, Plotly, Altair, etc.).

I am going to use the Altair library. It is a Python library for creating visualizations interactively and is easy to use. We can use this library to display bar charts, scatter plots, histograms, and many other charts. It is easy to customize charts using Altair, with different attributes represented in different colors, sizes, and styling options. These help us in making charts that can be dynamically changed by user inputs. We can integrate these charts with other Python libraries like pandas, NumPy, etc. This makes it easy to create charts with minimal coding.

When we can use other python libraries in the charts, it will help us in customizing the charts very quickly and easily, because we are using predefined libraries that reduces writing long lines of codes. There are also functions like scatter, bar etc., that can simply plot the chart by giving x and y variables.

In conclusion, It is an extremely powerful and flexible library, to design dynamic and interactive charts, using many predefined functions with minimal coding.

1.3 Create a selection object on your graph and bound it to one of the legends. Submit a screenshot of the graph which contains the selected object and a screenshot of your code (commented properly).

The chart shows the Sales and Price data for vehicles made by various Manufacturers. All the data points can be viewed for the dataset. We are using a dropdown to display all the unique manufacturers in the dataset. Then we use a selection function to enable a dropdown menu from where the users can dynamically select the manufacturer. Basically, we are binding the dropdown menu to the selection object.

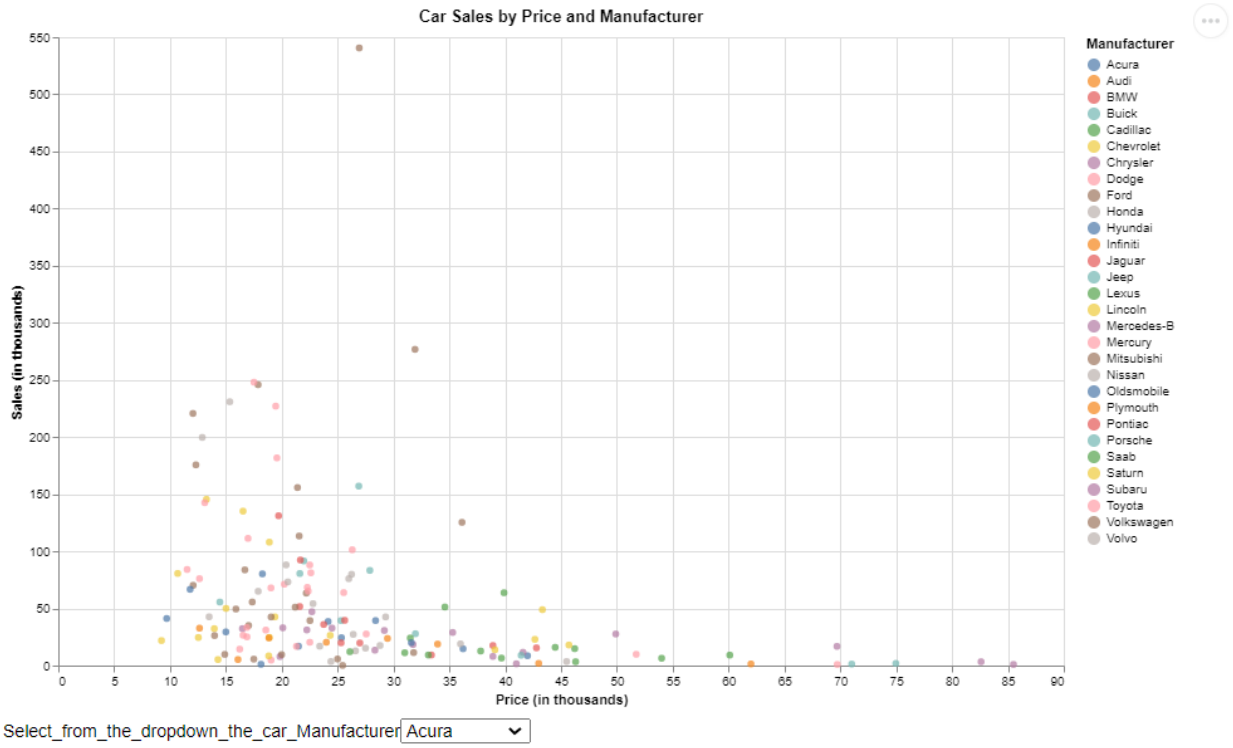
We are creating a scatter plot, with x and y axis encoded with Price_in_thousands and Sales_in_thousands columns of the car sales dataset, respectively. We are using color to encode the Manufacturer column, which displays data points with respect to each manufacturer. This is how we bound the manufacturer legend to the chart.

We use properties to indicate the size of the chart and we even display the title. Then we attach the selection object to the chart.

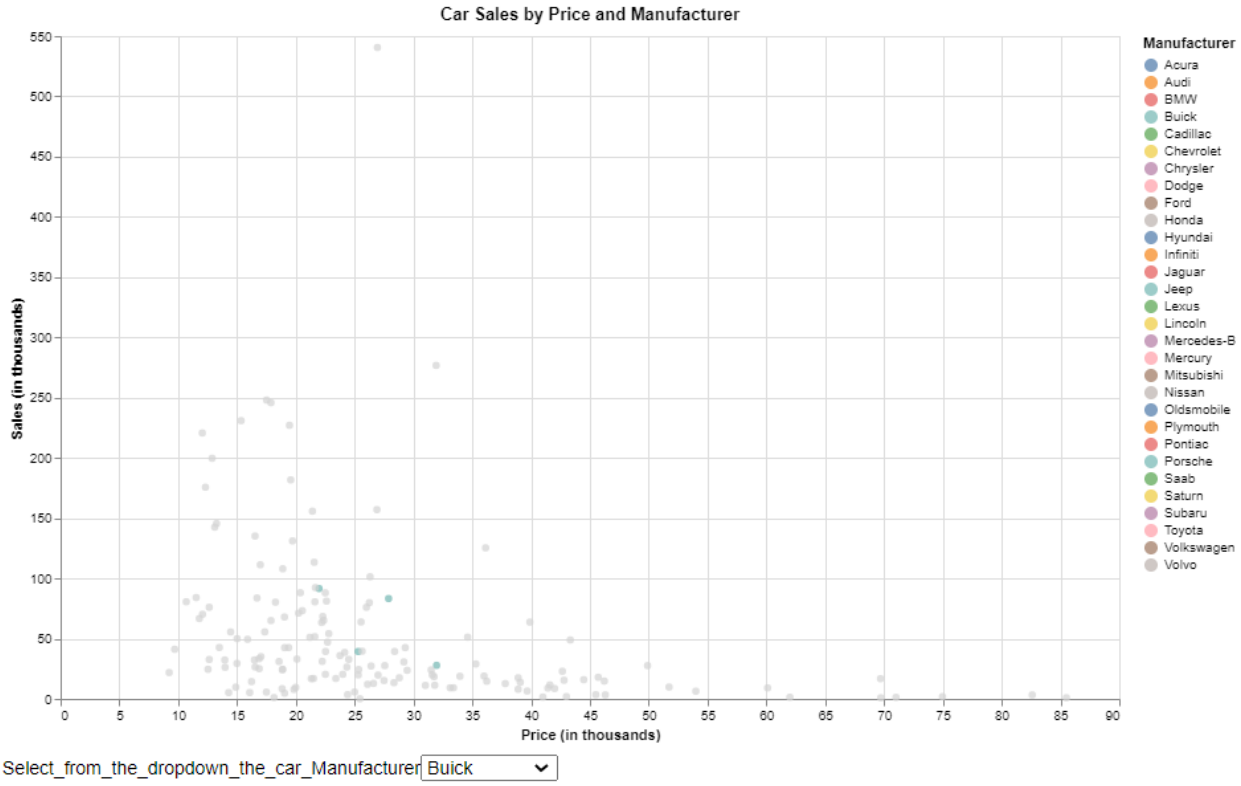
Finally, we are using the scatter to display the chart.

```
# Create the dropdown selection
dropdown = alt.binding_select(options=list(carsales['Manufacturer'].unique()))
selection = alt.selection_single(fields=['Manufacturer'], bind=dropdown, name='Select from the dropdown the car')
# Create the interactive selection to change scale of chart
interval = alt.selection_interval()
# Create the scatter plot
scatter = alt.Chart(carsales).mark_circle().encode(
    x=alt.X('Price_in_thousands:Q', title='Price (in thousands)'),
    y=alt.Y('Sales_in_thousands:Q', title='Sales (in thousands)'),
    color=alt.condition(selection, 'Manufacturer:N', alt.value('lightgray')),
).add_selection(selection).properties(
    width=800,
    height=500,
    title='Car Sales by Price and Manufacturer'
)

# Show the plot
scatter
```

The plot is presented below. It indicates Sales in relation to Vehicle Prices for each Manufacturer. Each of the colors stands for various Manufacturers. All manufacturer's colors are displayed in the legend. The titles and labels are present.



I have selected a Manufacturer Buick as shown below. Now we can see that all data points have turned gray but the data points for Buick are shown in blue color.



1.4 Create multiple plots which contain one interactive legend. Submit a screenshot of the multiple plots and a screenshot of your code (commented properly). Add a selection object on the multiple plots, and submit a screenshot of the selected multiple plots and a screenshot of your code (commented properly). Describe differences between multiple plots. Analysis of the data based on the plots.

Now we are creating multiple charts that are bound to a single legend.

We are creating two scatter plots; one shows the Sales and Price trends whereas the other shows the performance factor with respect to the price.

For the first chart, we encoded the x and y variables mapped to the Price_in_thousands and Sales_in_thousands columns of the car sales dataset, respectively. We are using color to encode the Manufacturer column, which displays data points with respect to each manufacturer. We use properties to indicate the size of the chart and we even display the title. Then we attach the selection object to the chart.

For the second chart, we encoded the x and y variables mapped to the Price_in_thousands and Power_perf_factor columns of the car sales dataset, respectively. We are using color to encode the Manufacturer column, which displays data points with respect to each

manufacturer. We use properties to indicate the size of the chart and we even display the title. Then we attach the selection object to the chart.

Finally, we are using hconcat function to attach both the scatter plots and the legends. We use the final chart to display the multiple plots bound to a single legend.

The first chart displays the trends in the Sales with respect to the price of the vehicle. The second chart shows how the performance of the vehicle changes with the price,

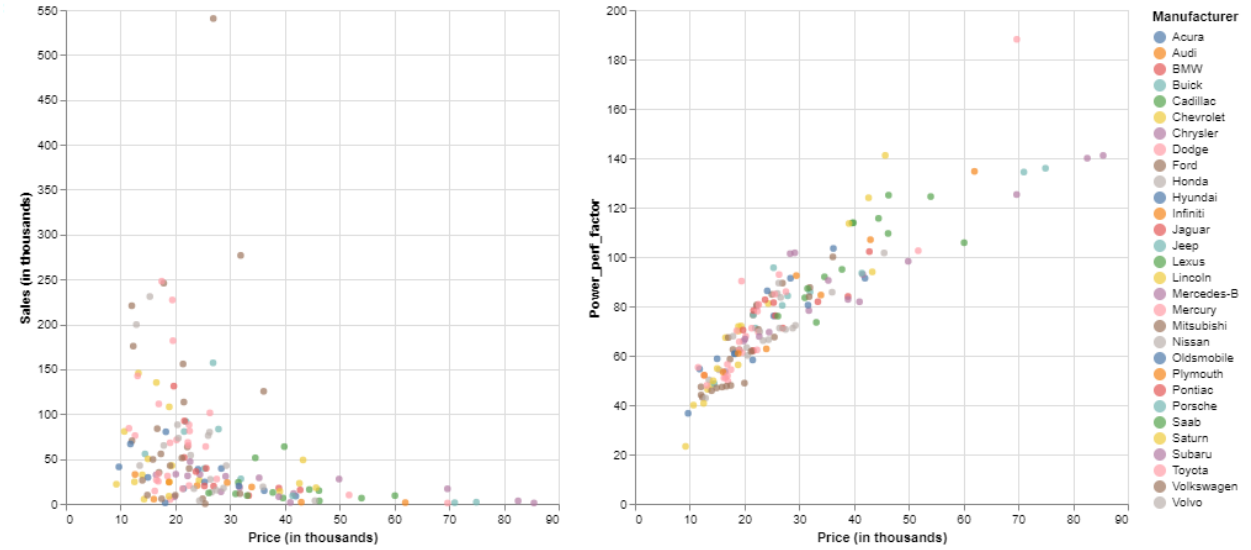
Interactive Legends

We are using legend to display all the unique manufacturers in the dataset. Then we use a selection function and bind it to the legend where the users can dynamically select the manufacturer.

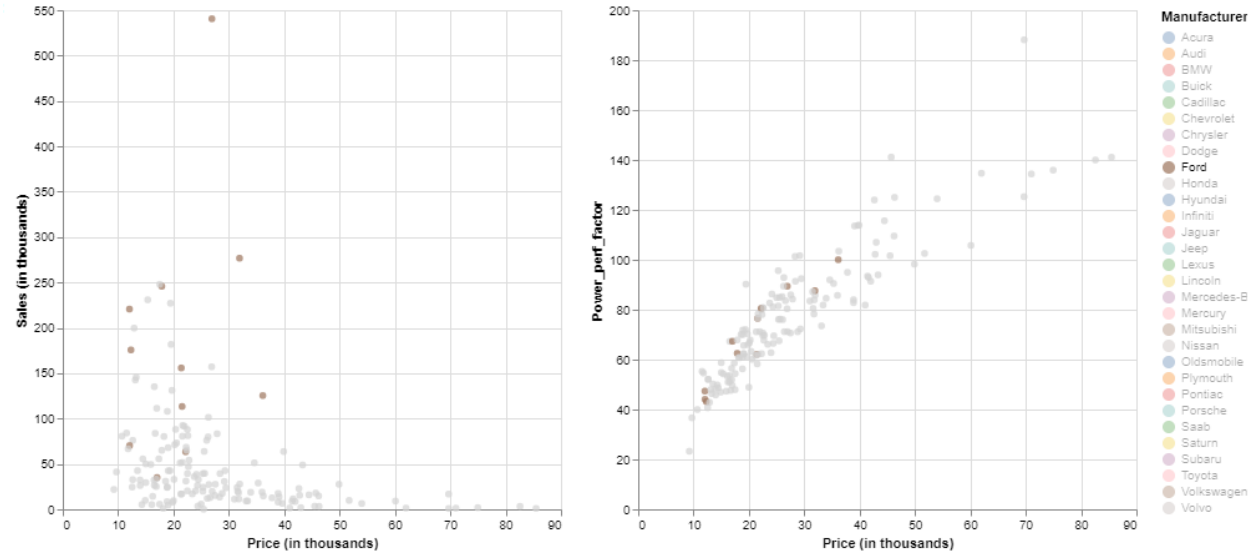
```
# Create the dropdown selection
dropdown = alt.binding_select(options=list(carsales['Manufacturer'].unique()))
selection = alt.selection_single(fields=['Manufacturer'], bind='legend', name='Select from the dropdown the car')
# Create the scatter plot
scatter1 = alt.Chart(carsales).mark_circle().encode(
    x=alt.X('Price_in_thousands:Q', title='Price (in thousands)'),
    y=alt.Y('Sales_in_thousands:Q', title='Sales (in thousands)'),
    color=alt.condition(selection, 'Manufacturer:N', alt.value('lightgray')),
    tooltip=['Manufacturer:N']
).add_selection(selection).properties(
    width=400,
    height=400
)


# Create the scatter plot
scatter2 = alt.Chart(carsales).mark_circle().encode(
    x=alt.X('Price_in_thousands:Q', title='Price (in thousands)'),
    y=alt.Y('Power_perf_factor:Q', title='Power_perf_factor'),
    color=alt.condition(selection, 'Manufacturer:N', alt.value('lightgray')),
    tooltip=['Model:N']
).add_selection(selection).properties(
    width=400,
    height=400
)



# Combine plots and legend into a single chart
charts = alt.hconcat(scatter1, scatter2)

chart_with_legend = charts

# Display chart
chart_with_legend
```

In the first chart we can observe that the major datapoints are located between the 10 to 45-thousand-dollar Price and between the 0-150 thousand sales. We can see that the sales decreased with the increase in the price with respect to all manufacturers. Each manufacturer is displayed using a different color.

In the second chart, we can see that the performance of the vehicles is high when the price is high. Majority of the vehicle in the price range of 10-40 thousand dollars have a performance factor between the range 40-100. The performance is low for vehicles under the 10,000-price range.



Below are the multiple plots with interactive legend selected, Here I have selected Ford manufacturer. As you can see the data points with respect to the manufacturer Ford are highlighted in both the charts. They appear brown in color.



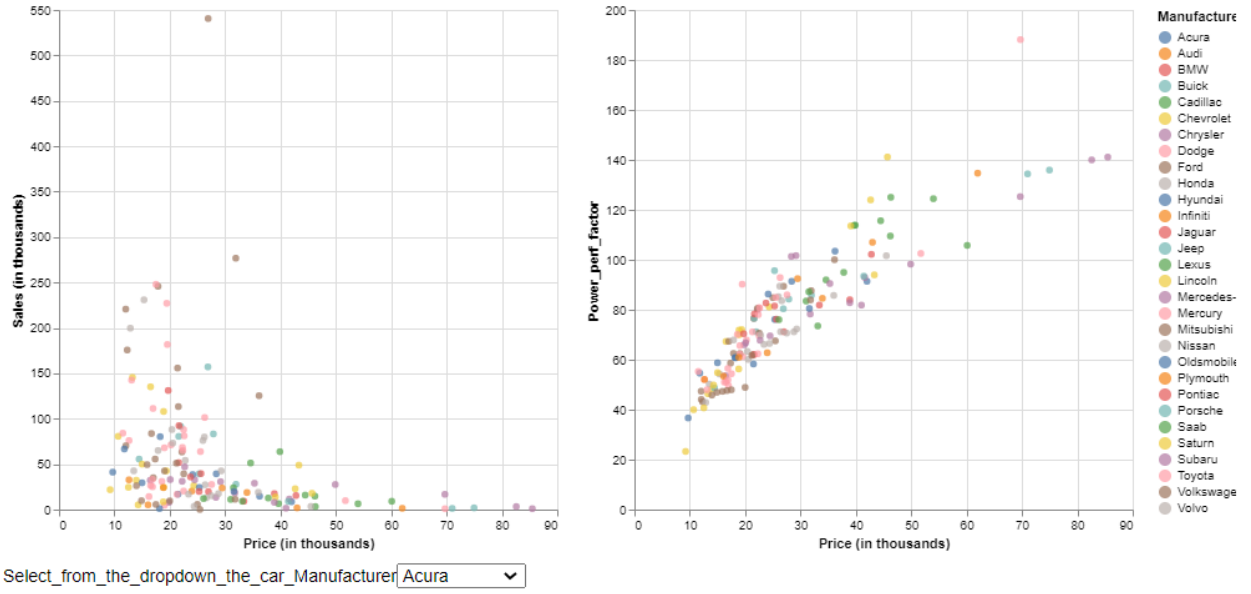
Selection Object

We are using a dropdown to display all the unique manufacturers in the dataset. Then we use a selection function to enable a dropdown menu from where the users can dynamically select the manufacturer.

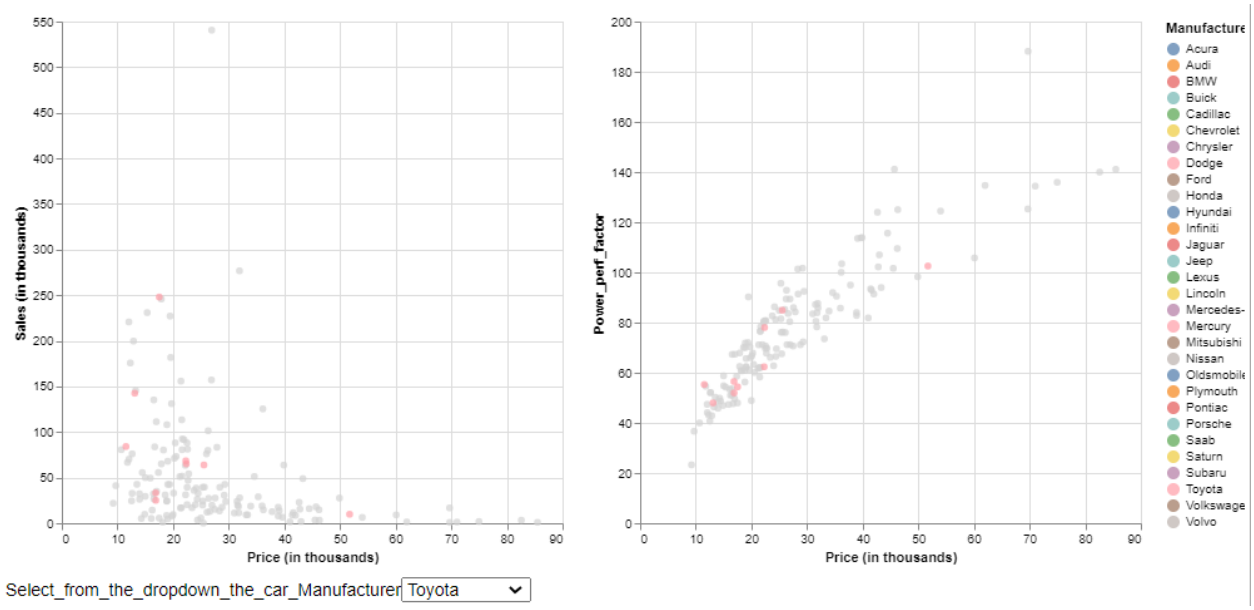
```
# Create the dropdown selection
dropdown = alt.binding_select(options=list(carsales['Manufacturer'].unique()))
selection = alt.selection_single(fields=['Manufacturer'], bind=dropdown, name='Select from the dropdown the car')
# Create the scatter plot
scatter1 = alt.Chart(carsales).mark_circle().encode(
    x=alt.X('Price_in_thousands:Q', title='Price (in thousands)'),
    y=alt.Y('Sales_in_thousands:Q', title='Sales (in thousands)'),
    color=alt.condition(selection, 'Manufacturer:N', alt.value('lightgray')),
    tooltip=['Manufacturer:N']
).add_selection(selection).properties(
    width=400,
    height=400
)


# Create the scatter plot
scatter2 = alt.Chart(carsales).mark_circle().encode(
    x=alt.X('Price_in_thousands:Q', title='Price (in thousands)'),
    y=alt.Y('Power_perf_factor:Q', title='Power_perf_factor'),
    color=alt.condition(selection, 'Manufacturer:N', alt.value('lightgray')),
    tooltip=['Model:N']
).add_selection(selection).properties(
    width=400,
    height=400
)



# Combine plots and legend into a single chart
charts = alt.hconcat(scatter1, scatter2)

chart_with_legend = charts

# Display chart
chart_with_legend
```

In the first chart we can observe that the major datapoints are located between the 10 to 45-thousand-dollar Price and between the 0-150 thousand sales. We can see that the sales decreased with the increase in the price with respect to all manufacturers. Each manufacturer is displayed using a different color.

In the second chart, we can see that the performance of the vehicles is high when the price is high. Majority of the vehicle in the price range of 10-40 thousand dollars have a performance factor between the range 40-100. The performance is low for vehicles under the 10,000-price range.



Below are the multiple plots with selection object, Here I have selected Toyota manufacturer. As you can see the datapoints with respect to the manufacturer Toyota are highlighted in both the charts. They appear in pink color.



II. PANNING AND ZOOMING

The chart shows the Sales and Price data for vehicles made by various Manufacturers. All the data points can be viewed for the dataset. We are using a dropdown to display all the unique manufacturers in the dataset. Then we use a selection function to enable a dropdown menu from where the users can dynamically select the manufacturer.

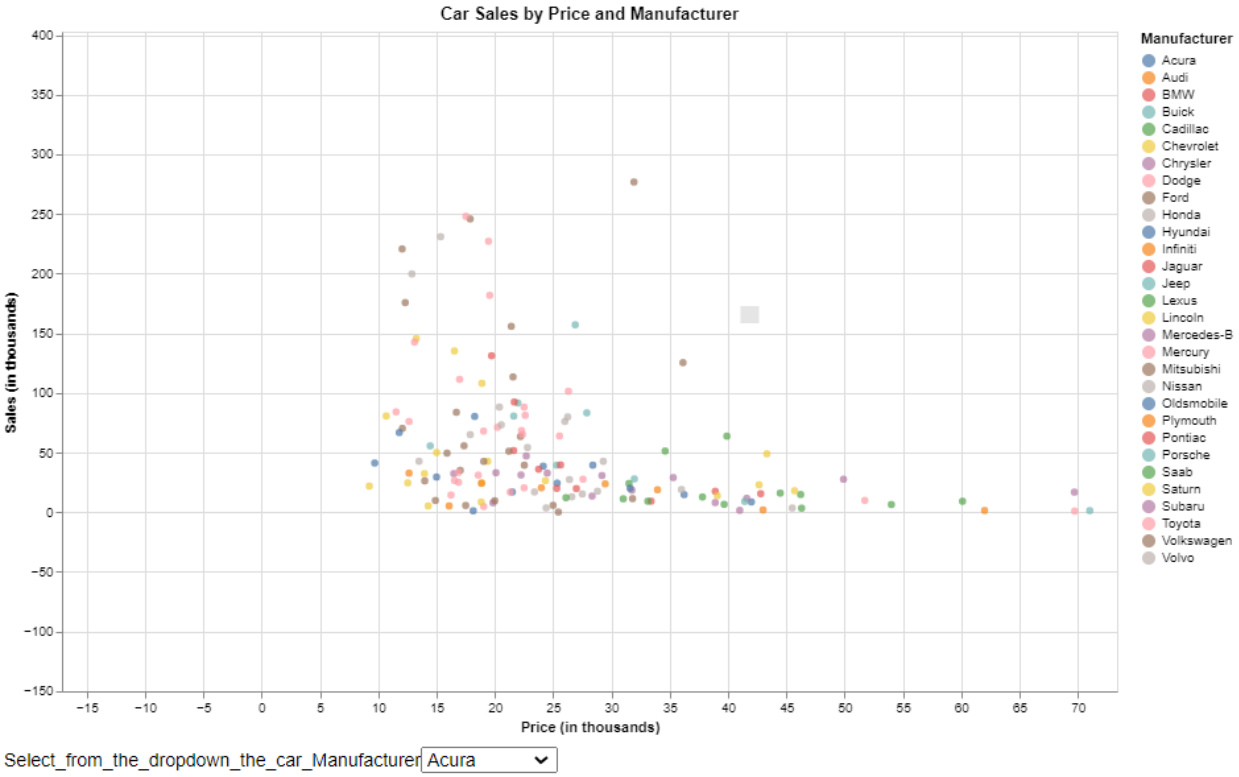
We are creating a scatter plot, we encoded the x and y variables to Price_in_thousands and Sales_in_thousands columns of the car sales dataset, respectively. We are using color to encode the Manufacturer column, which displays data points with respect to each manufacturer. This is how we bound the manufacturer legend to the chart.

2.1 Panning on the graph. Submit a screenshot of the graph and a screenshot of your code (commented properly).

We are also using the selection_interval function from panning across the x and y axis of the graph. We use properties to indicate the size of the chart and we even display the title. Then we attach the selection object to the chart. We also added an interactive function to the chart. Finally, we are using the scatter to display the chart. This will make the chart spannable.

```
# Create the dropdown selection
dropdown = alt.binding_select(options=list(carsales['Manufacturer'].unique()))
selection = alt.selection_single(fields=['Manufacturer'], bind=dropdown, name='Select from the dropdown the car')
# Create the interactive selection to change scale of chart
interval = alt.selection_interval()
panning = alt.selection_interval(bind='scales', encodings=['x', 'y'])

# Create the scatter plot
scatter = alt.Chart(carsales).mark_circle().encode(
    x=alt.X('Price_in_thousands:Q', title='Price (in thousands)'),
    y=alt.Y('Sales_in_thousands:Q', title='Sales (in thousands)'),
    color=alt.condition(selection, 'Manufacturer:N', alt.value('lightgray')),
    tooltip=['Model:N']
).add_selection(selection).properties(
    width=800,
    height=500,
    title='Car Sales by Price and Manufacturer'
).add_selection(
    panning,interval
).interactive(bind_y=False)

# Show the plot
scatter
```

As you can see below, we can pan across the x and y axis. We can see the negative axes, that we did not see in the above charts.

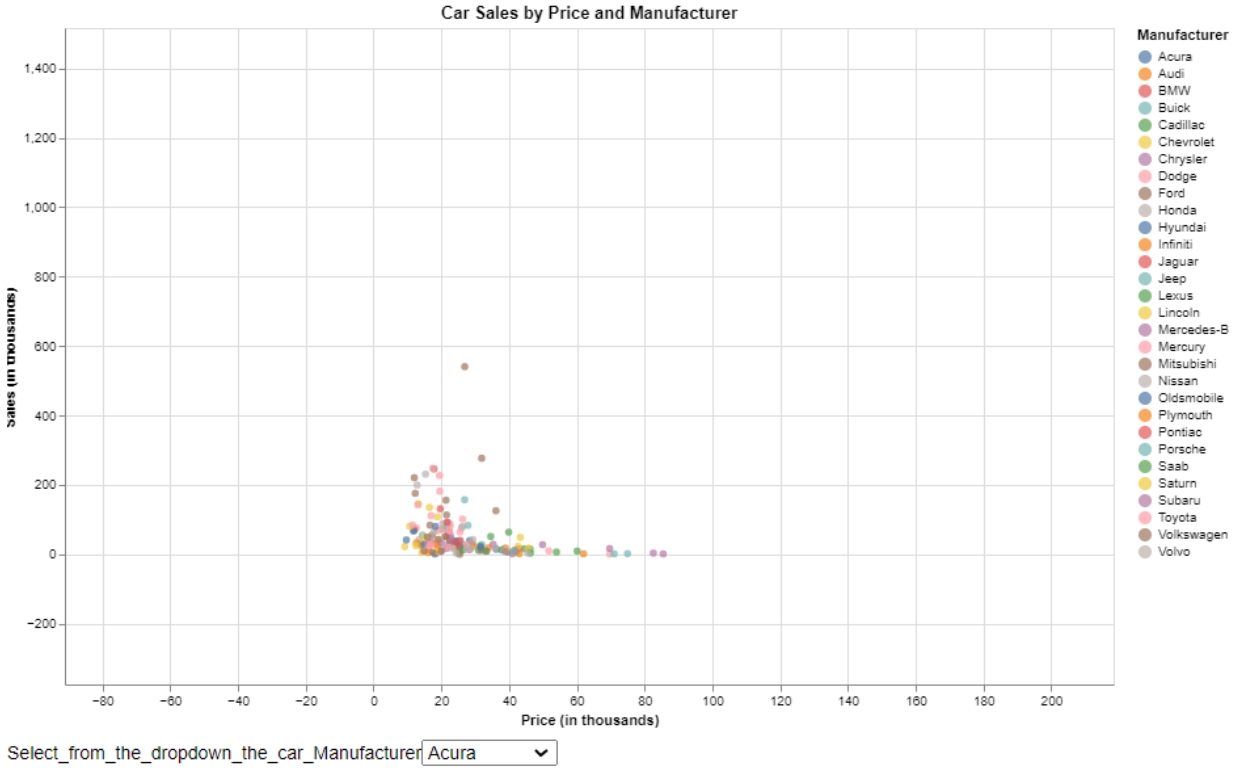


2.2 Zoom in and out on the graph. Submit two screenshots of the graph and a screenshot of your code (commented properly).

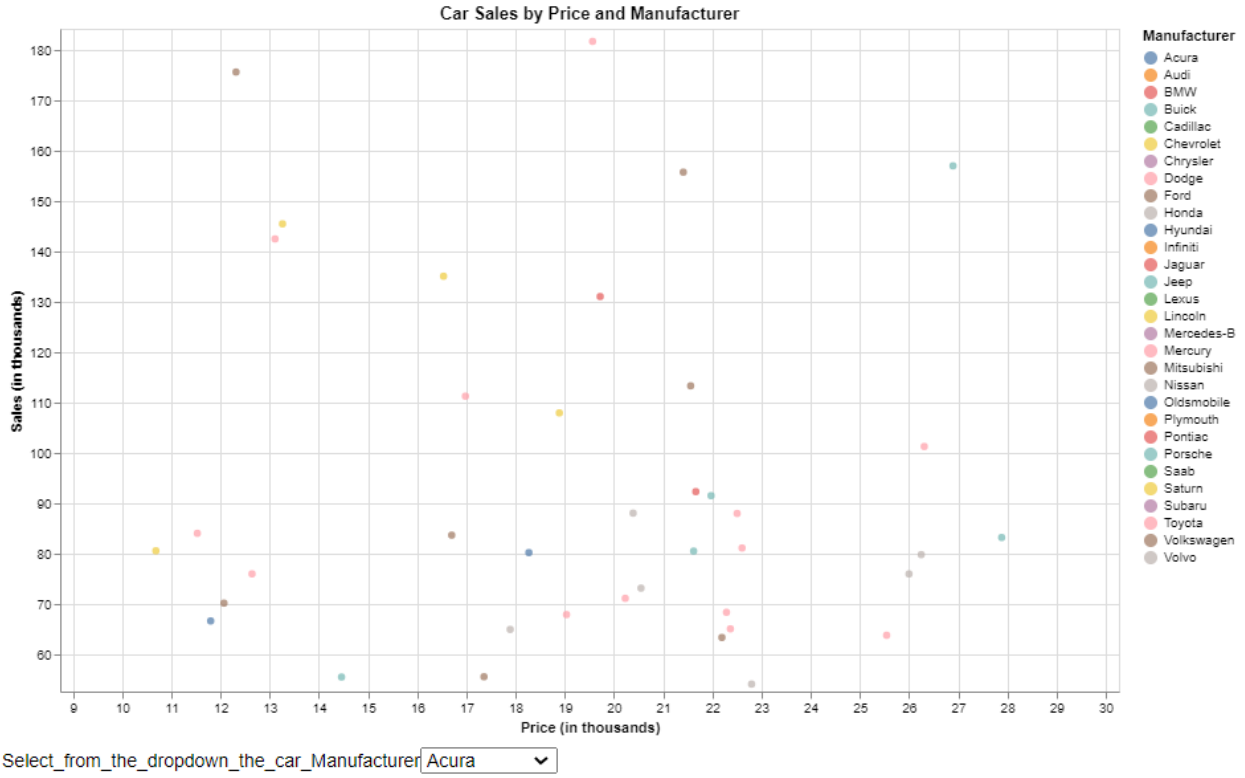
We are also using the selection_interval function and interactive function on the chart. We use properties to indicate the size of the chart and we even display the title. Then we attach the selection object to the chart. Finally, we are using the scatter to display the chart. This will make the chart zoomable.

```
# Create the dropdown selection
dropdown = alt.binding_select(options=list(carsales['Manufacturer'].unique()))
selection = alt.selection_single(fields=['Manufacturer'], bind=dropdown, name='Select from the dropdown the car')
# Create the interactive selection to change scale of chart
interval = alt.selection_interval()
# Create the scatter plot
scatter = alt.Chart(carsales).mark_circle().encode(
    x=alt.X('Price_in_thousands:Q', title='Price (in thousands)'),
    y=alt.Y('Sales_in_thousands:Q', title='Sales (in thousands)'),
    color=alt.condition(selection, 'Manufacturer:N', alt.value('lightgray')),
    tooltip=['Model:N']
).add_selection(selection).properties(
    width=800,
    height=500,
    title='Car Sales by Price and Manufacturer'
).interactive()

# Show the plot
scatter
```

To the below chart, we can zoom as we need. We can see that previous charts had 10 scales, now it is 20 scale graphs. Showing a minimized version of the graph.



To the below chart, we can zoom as we need. We can see that previous charts had 10 scales, now it is 1 scale graphs. Showing a maximized version of the graph.



2.3 What are the pros and cons of Panning and Zooming? Give sufficient explanation with examples.

Panning:

Pros:

- It helps the user to move across the axis for better visualization.
- It is useful when the data is distributed across and is wider than the figure size that we are displaying the chart in.
- We cannot use huge figures to display large datasets, instead we can add panning, that will help users to move across the scale and see all the data.

Example: For a scatter plot of price v/s sales across each year, we can simply use a chart and pan across the scale.

Cons:

- It might not be suitable for all kinds of data.
- When we are using charts to find trends in the data, panning can degrade the purpose of the charts.

Example: For a scatter plot of data to find trends in medical science across each year, we cannot pan and see all the trends.

Zooming:

Pros:

- Sometimes when the data is extremely dense, it is easy to zoom in and easily understand and analyze the data points.
- We can zoom out to find the trends that may not appear on the normal scale.
- We can identify internal correlations and outliers by zooming in and out of the charts.

Example: We can use zoom for time series data, where there are many datapoints and zooming can help better understand data.

Cons:

- It might not be suited for all kinds of data.
- Sometimes we need to just see the overview of the data, not any specific outliers or correlations.

Example: we do not need zoom for simple charts, that do not require any zooming, for example pie charts, etc.

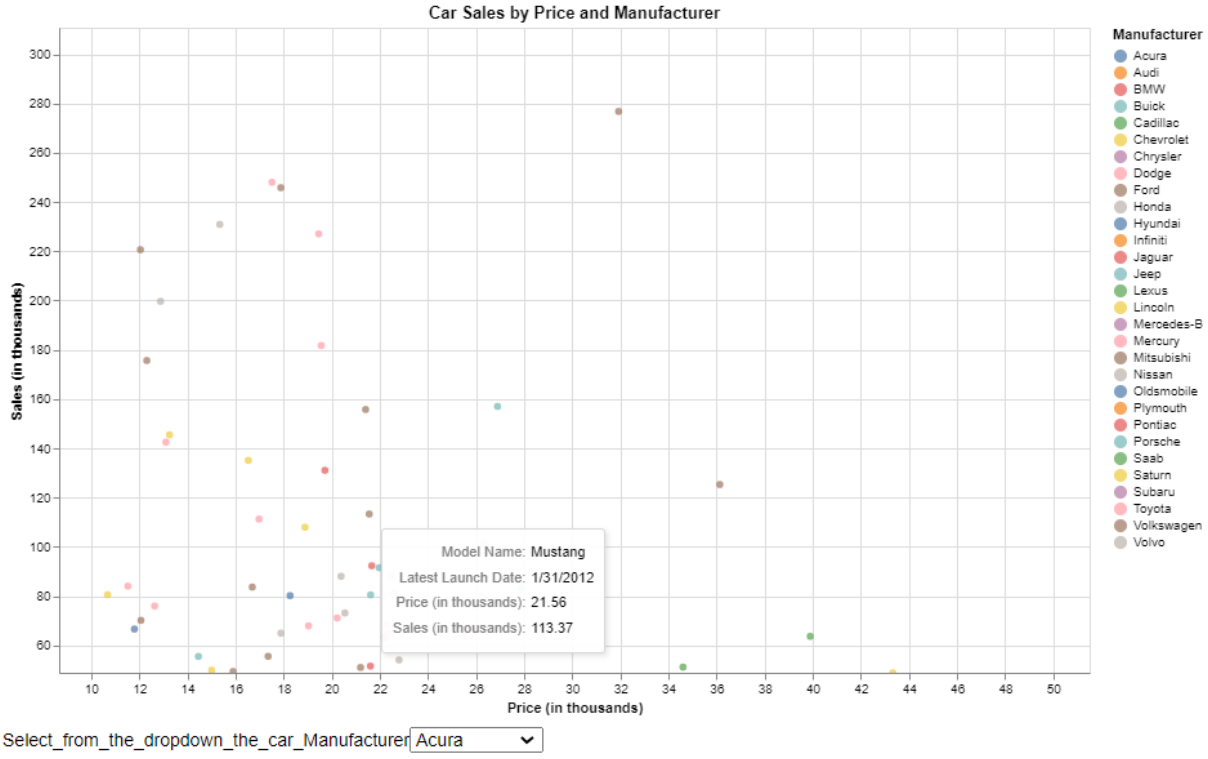
III. ADDING TOOLTIPS

3.1 Adding at least two different tooltips on your graph. Submit a screenshot of the graph and a screenshot of your code (commented properly).

To the above chart that was created in the previous section, we have now added a tooltip. We are using the tooltip function in the Altair library; this will help in adding multiple tootlips with customized title for the tooltips. These are nothing but an additional way to describe the data, to help users understand each datapoint and the category that they belong to.

```
# Create the scatter plot
scatter = alt.Chart(carsales).mark_circle().encode(
    x=alt.X('Price_in_thousands:Q', title='Price (in thousands)'),
    y=alt.Y('Sales_in_thousands:Q', title='Sales (in thousands)'),
    color=alt.condition(selection, 'Manufacturer:N', alt.value('lightgray')),
    tooltip=[alt.Tooltip('Model:N', title='Model Name'),
             alt.Tooltip('Latest_Launch:N', title='Latest Launch Date'),
             alt.Tooltip('Price_in_thousands:Q', title='Price (in thousands)', format='.2f'),
             alt.Tooltip('Sales_in_thousands:Q', title='Sales (in thousands)', format='.2f')]
).add_selection(selection).properties(
    width=800,
    height=500,
    title='Car Sales by Price and Manufacturer'
).interactive()

# Show the plot
scatter
```

As you can see below the data point shows four values in the tooltip. It shows the Model Name, Latetst Launch Date, Price and Sales. All the data points display this tooltip, making it easy for the user to interactively get more information about the data.



3.2 Why you are choosing these elements/ labels as tooltips. What are the advantages with or without the tooltips?

This chart shows trends in sales with respect to the price of vehicle. I have used Model Name, Latetst Launch Date, Price and Sales for tootlip. I am using these elements because the provide more inormation about each vehicle. Though we are color coding the Manufacturer, it might not be enough in some cases, Adding tootltip helps user to even check which model has high sales for a particlar Manufacturer, we do not have to plot another chart for this, simply adding a tooltip is enough to understand and analyze.

Advantages with tooltip:

- Adding a tooltip presents more information about the data, that can help the users to understand the data better.
- By adding a tooltip, we can avoid creating redundant charts. Instead we can use provide more information using a tooltip.
- It is also useful in proving specific information, be it a histogram or a scatter plot or any other chart.

Advantages without tooltip:

- Sometimes, when we are using simple charts that do not require extra data, tooltips are just additional lines of code in that case. It would be an advantage to not use a tooltip in that case.
- Sometimes tooltips can give unnecessary information, that can distract the actual purpose of the charts.

IV. SCATTER PLOT WITH HREF

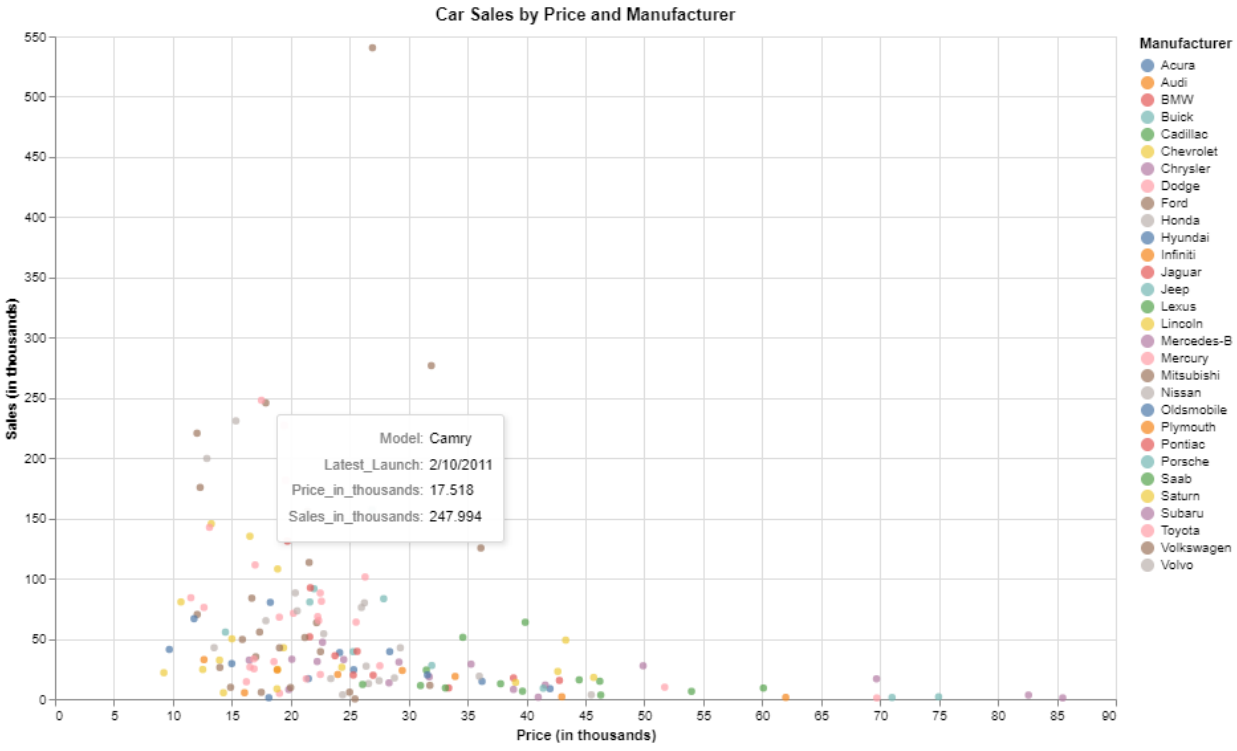
4.1 Consider the Task-1 graph(1.1 standard Scatter plot), add a tooltip to it. submit the screenshot of the graph and a screenshot of your code (commented properly).

I am plotting a scatter plot using my Car Sales dataset. To visualize my data and create an interactive chart, using the Altair library. I am using only the Manufacturer, Sales_in_thousands, and Price_in_thousands columns to show the relationship between car price and sales, grouped by manufacturer. We have also added a tooltip to display Model Name, Latetst Launch Date, Price and Sales.

Now to the scatter chart, we have encoded the x and y variables that are mapped to the Price_in_thousands and Sales_in_thousands columns of the car sales dataset, respectively. We are using color to encode the Manufacturer column, which displays data points with respect to each manufacturer. We use properties to indicate the size of the chart and we even display the title. Finally, we are using the scatter to display the chart.

```
# Create the scatter plot
scatter = alt.Chart(carsales).mark_circle().encode(
    x=alt.X('Price_in_thousands:Q', title='Price (in thousands)'),
    y=alt.Y('Sales_in_thousands:Q', title='Sales (in thousands)'),
    color=alt.Color('Manufacturer', title='Manufacturer'),
    tooltip=['Model:N', 'Latest_Launch:N', 'Price_in_thousands:Q', 'Sales_in_thousands:Q']
).add_selection().properties(
    width=800,
    height=500,
    title='Car Sales by Price and Manufacturer'
).add_selection().properties(
    width=800,
    height=500,
    title='Car Sales by Price and Manufacturer'
)

# Display the scatter plot
scatter
```

As you can see below, we have added a tooltip to display Model Name, Latetst Launch Date, Price and Sales.

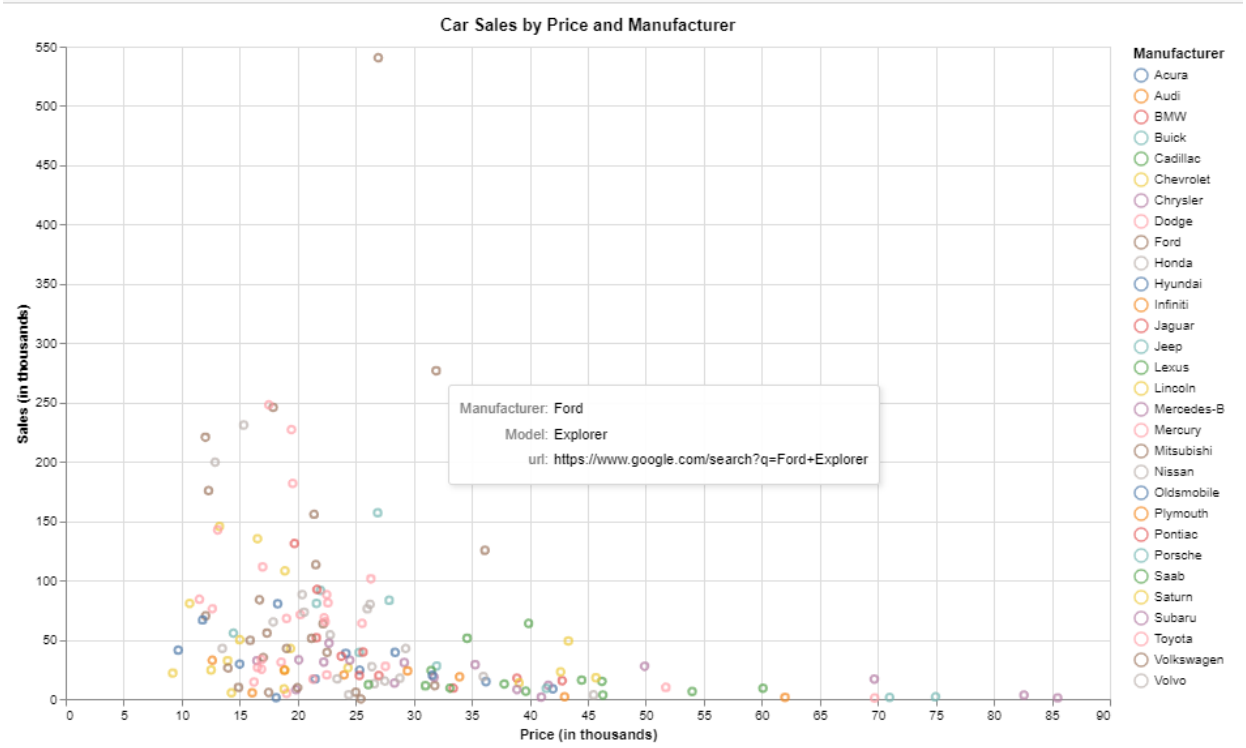


4.2 Add href to your graph and add URL to tooltip, when you click on any of the points it must open a corresponding google search. Submit the screenshot of your code (commented properly) and submit the screenshot of the graph showing tooltips and the redirected website.

To the above code I have added a Href variable that uses a google URL and appends it with the manufacturer name and the model's name. We are also displaying this URL in the tooltip. When we click on the datapoint, it will redirect us to the Google page of the Manufacturer and the particular model that has been selected.

```
# Create the scatter plot with href
alt.Chart(carsales).transform_calculate(
    url='https://www.google.com/search?q=' + alt.datum.Manufacturer + '+' + alt.datum.Model
).mark_point().encode(
    x=alt.X('Price_in_thousands:Q', title='Price (in thousands)'),
    y=alt.Y('Sales_in_thousands:Q', title='Sales (in thousands)'),
    color=alt.Color('Manufacturer', title='Manufacturer'),
    href='url:N',
    tooltip=['Manufacturer:N', 'Model:N', 'url:N']
).add_selection().properties(
    width=800,
    height=500,
    title='Car Sales by Price and Manufacturer'
)
```

As you can see below, the datapoint we selected is a Ford Explorer. Now we click on it.



The webpage is opened as shown below.

Ford Explorer

Images | Shopping | News | Videos | Maps | Books | Flights | Finance | Tools

About 116,000,000 results (0.51 seconds)

Ford
https://www.ford.com › suvs › explorer

2023 Ford Explorer SUV USA | Pricing, Photos, Specs

The 2023 **Ford Explorer** SUV is available in 8 trim levels with 3 engine options, including hybrid, in the USA. Enjoy seating for up to 7 & plenty of cargo ...

2023 Ford Explorer SUV

The 2023 Ford Explorer SUV is available in 8 trims including ...

2023 Explorer XLT

... transmission & Terrain Management System™ on the ...

Explorer Limited

The 2023 Ford Explorer Limited SUV comes with a 2.3L ...

2022 Ford Explorer

The 2022 Ford Explorer SUV has seating for up to 7 & Intelligent ...

More results from ford.com »

2023 Ford Explorer

SUV

7.5/10	7.7/10	7.9/10
Car and Driver	Edmunds	MotorTrend