In [117]:

```
import pandas as pd
import altair as alt
```

In [118]:

```
# Load the data
carsales = pd.read_csv('https://gist.githubusercontent.com/nehabaddam/1f47243b
carsales.head()
```

Out[118]:

	Manufacturer	Model	Sales_in_thousands	__year_resale_value	Vehicle_type	Price_in_thousai
0	Acura	Integra	16.919	16.360	Passenger	21
1	Acura	TL	39.384	19.875	Passenger	28
2	Acura	CL	14.114	18.225	Passenger	N
3	Acura	RL	8.588	29.725	Passenger	42
4	Audi	A4	20.397	22.255	Passenger	23

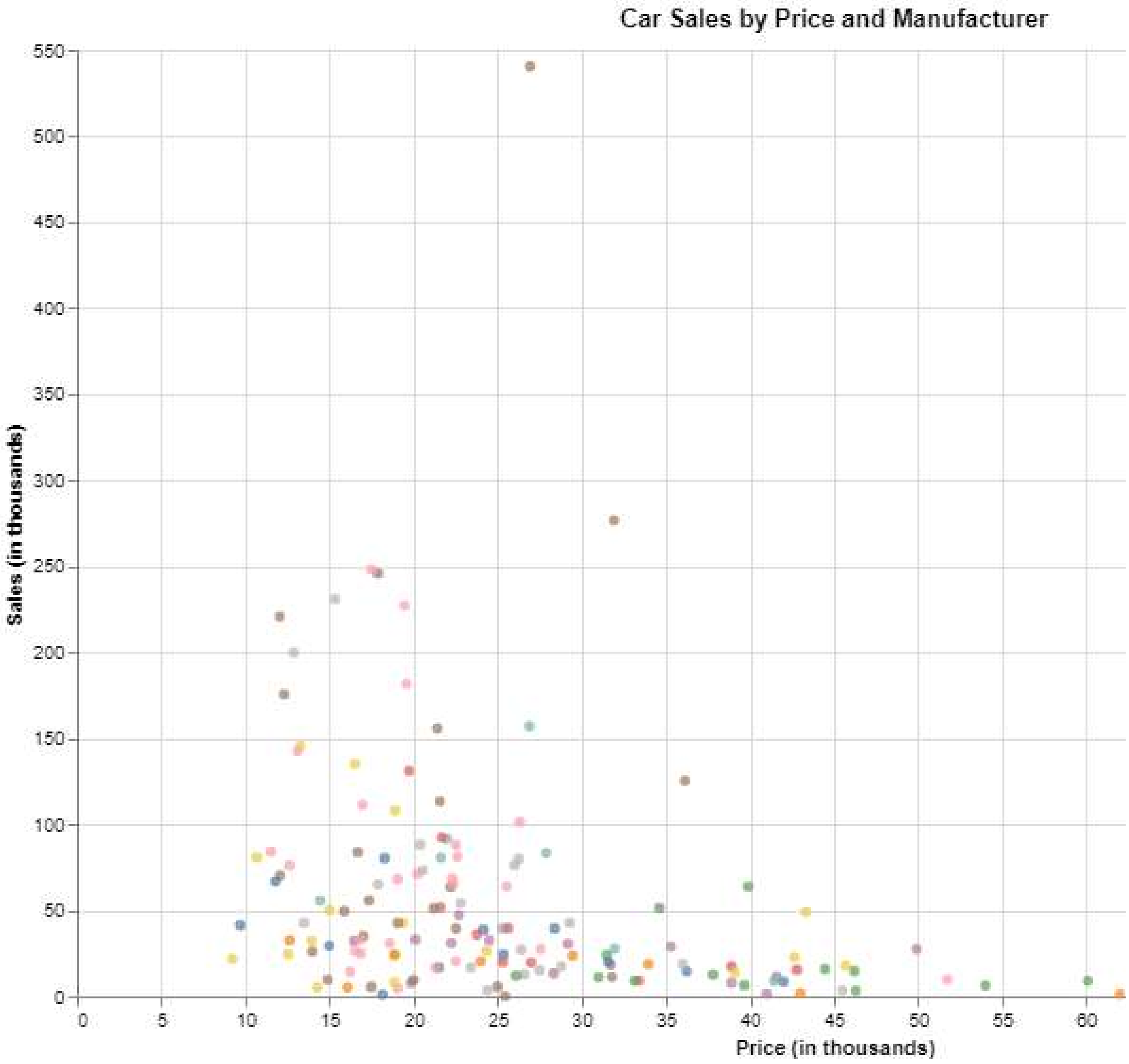
1. MAKING INTERACTIVE VISUALIZATIONS

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

In [119]:

```python
# 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
```

Out[119]:



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.).

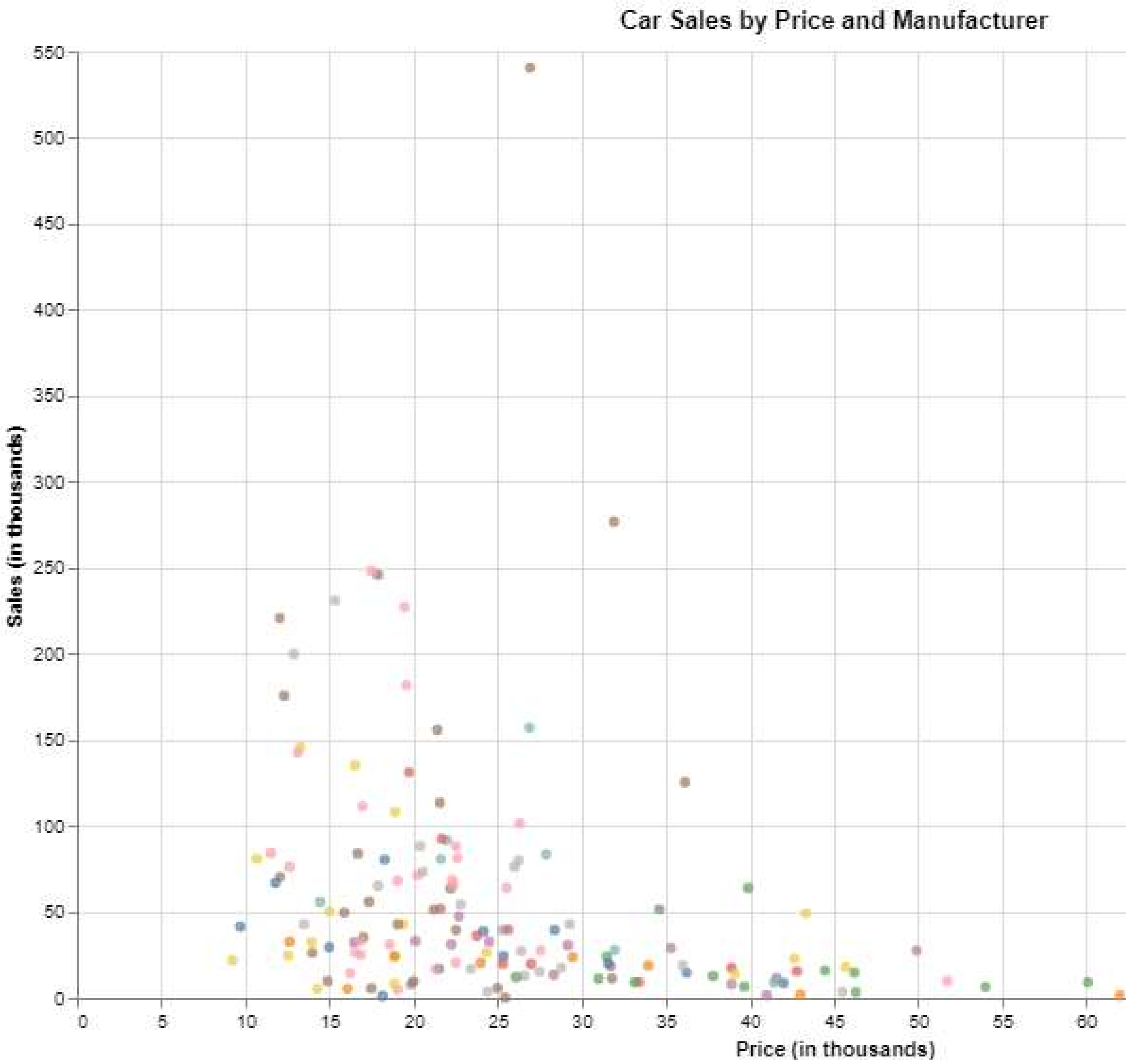
I am going to use both Plotly and Altair

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 selection object and a screenshot of your code (commented properly).

In [120]:

```
# Create the dropdown selection
dropdown = alt.binding_select(options=list(carsales['Manufacturer'].unique()))
selection = alt.selection_single(fields=['Manufacturer'], bind=dropdown, name=
# 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
```

Out[120]:



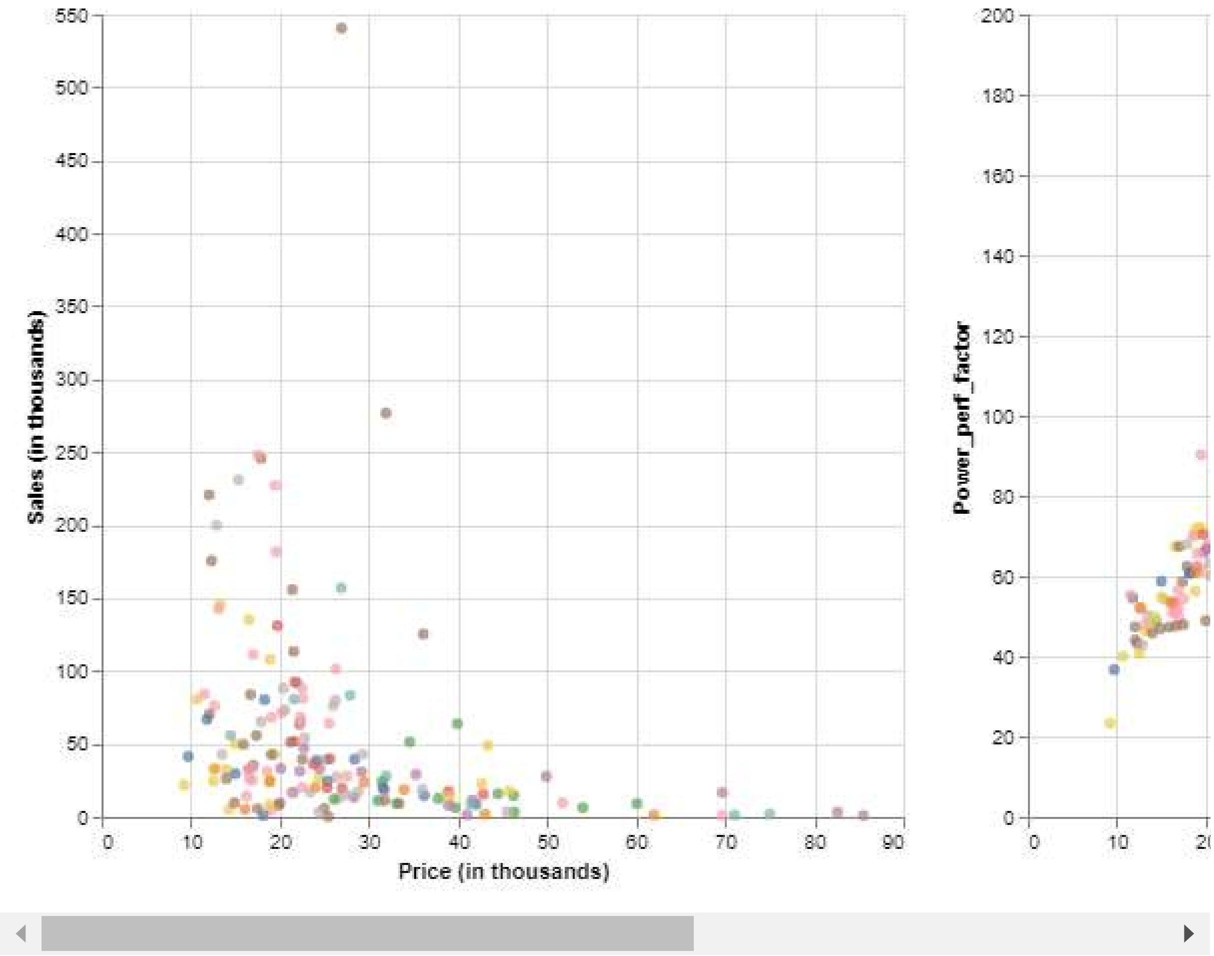
Select_from_the_dropdown_the_car_Manufacturer Acura ⌄

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).

In [123]:

```
# Create the dropdown selection
dropdown = alt.binding_select(options=list(carsales['Manufacturer'].unique()))
selection = alt.selection_single(fields=['Manufacturer'], bind='legend', name=
# 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
```

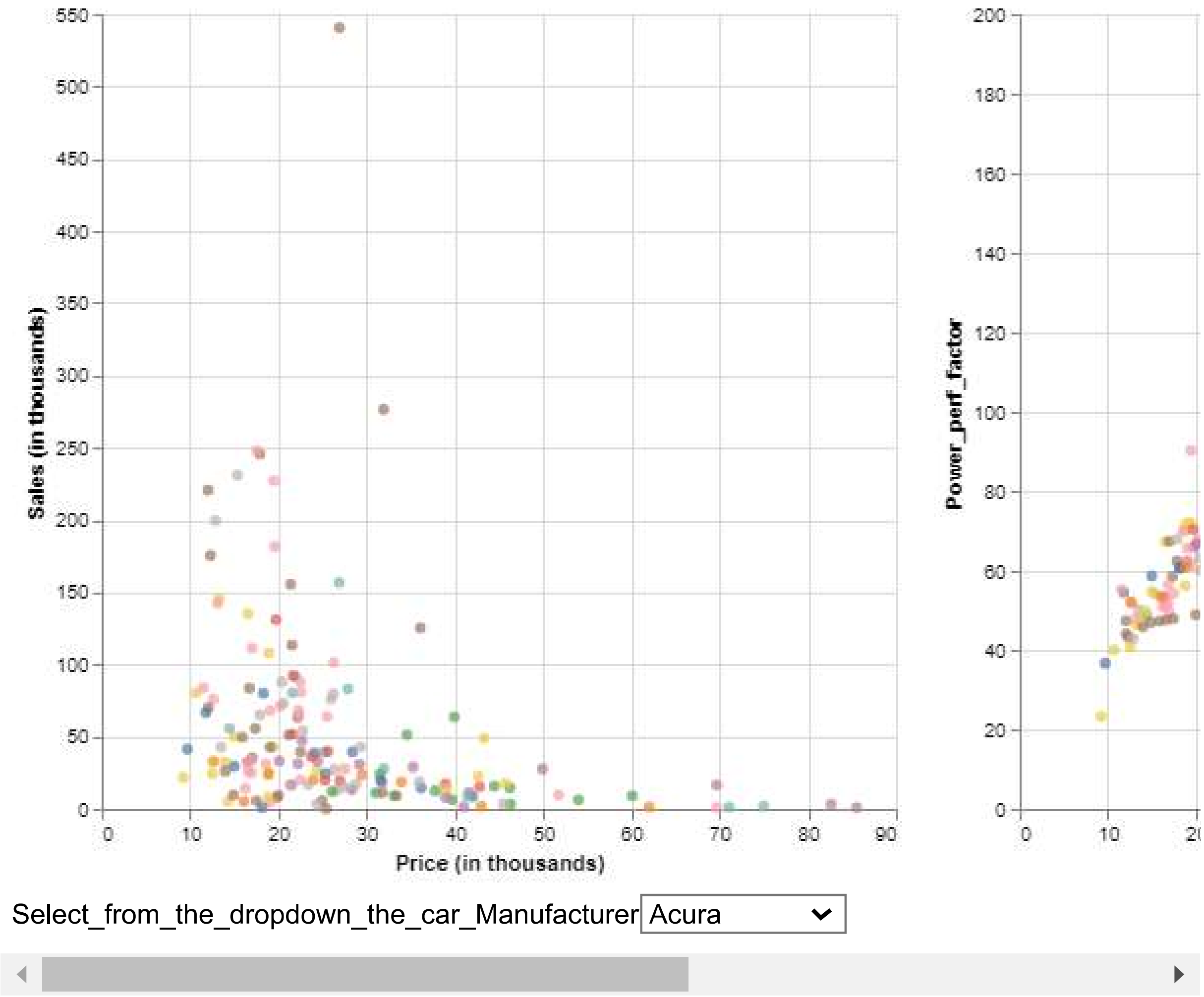
Out[123]:


Sales (in thousands)
550
500
450
400
350
300
250
200
150
100
50
0
0
10
20
30
40
50
60
70
80
90
Price (in thousands)
Power_perf_factor
200
180
160
140
120
100
80
60
40
20
0
0
10
2(

In [124]:

```
# Create the dropdown selection
dropdown = alt.binding_select(options=list(carsales['Manufacturer'].unique()))
selection = alt.selection_single(fields=['Manufacturer'], bind=dropdown, name=
# 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
```

Out[124]:



Select_from_the_dropdown_the_car_Manufacturer Acura

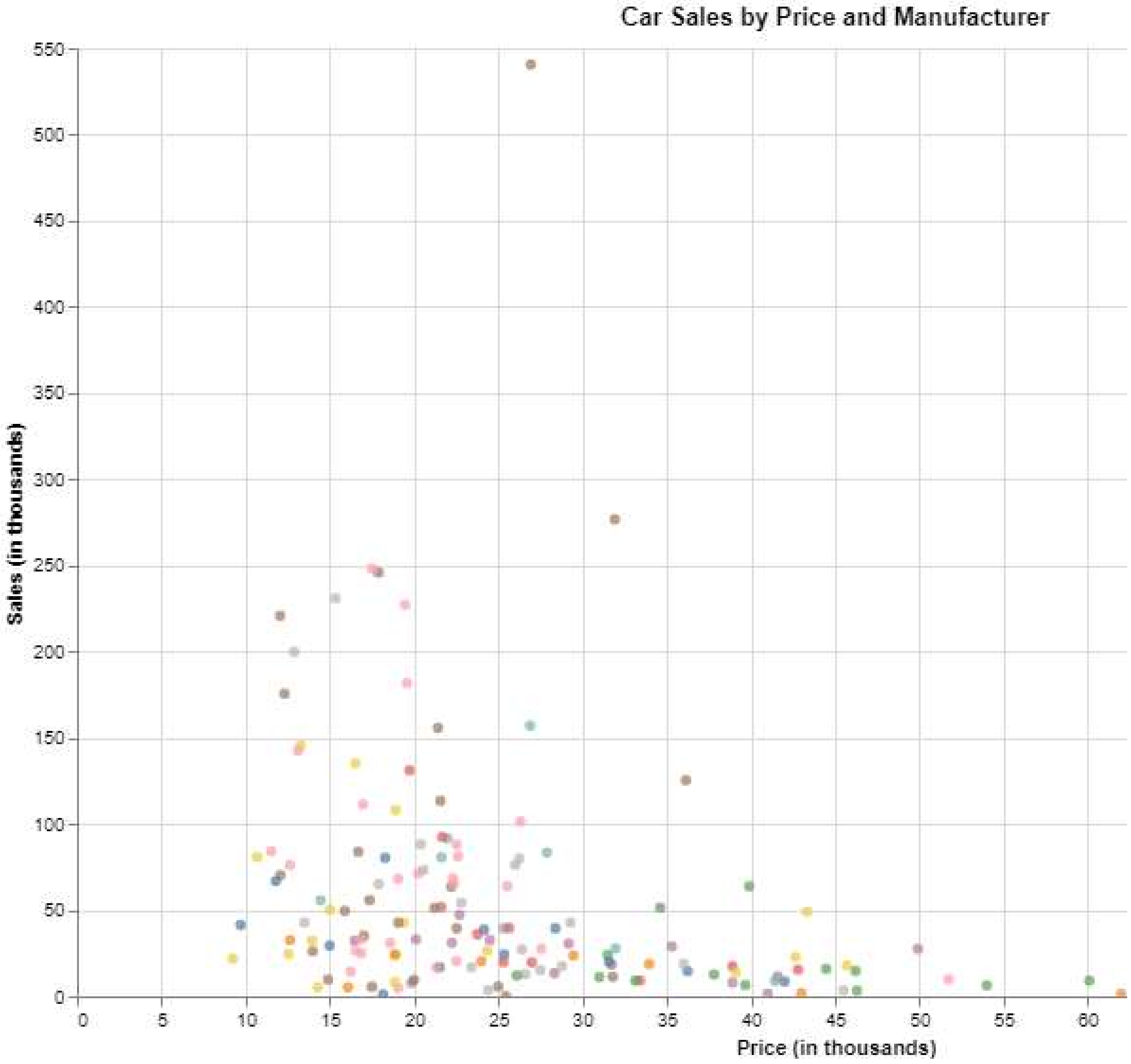
2. PANNING AND ZOOMING

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

In [126]:

```
# Create the dropdown selection
dropdown = alt.binding_select(options=list(carsales['Manufacturer'].unique()))
selection = alt.selection_single(fields=['Manufacturer'], bind=dropdown, name=
# 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
```

Out[126]:



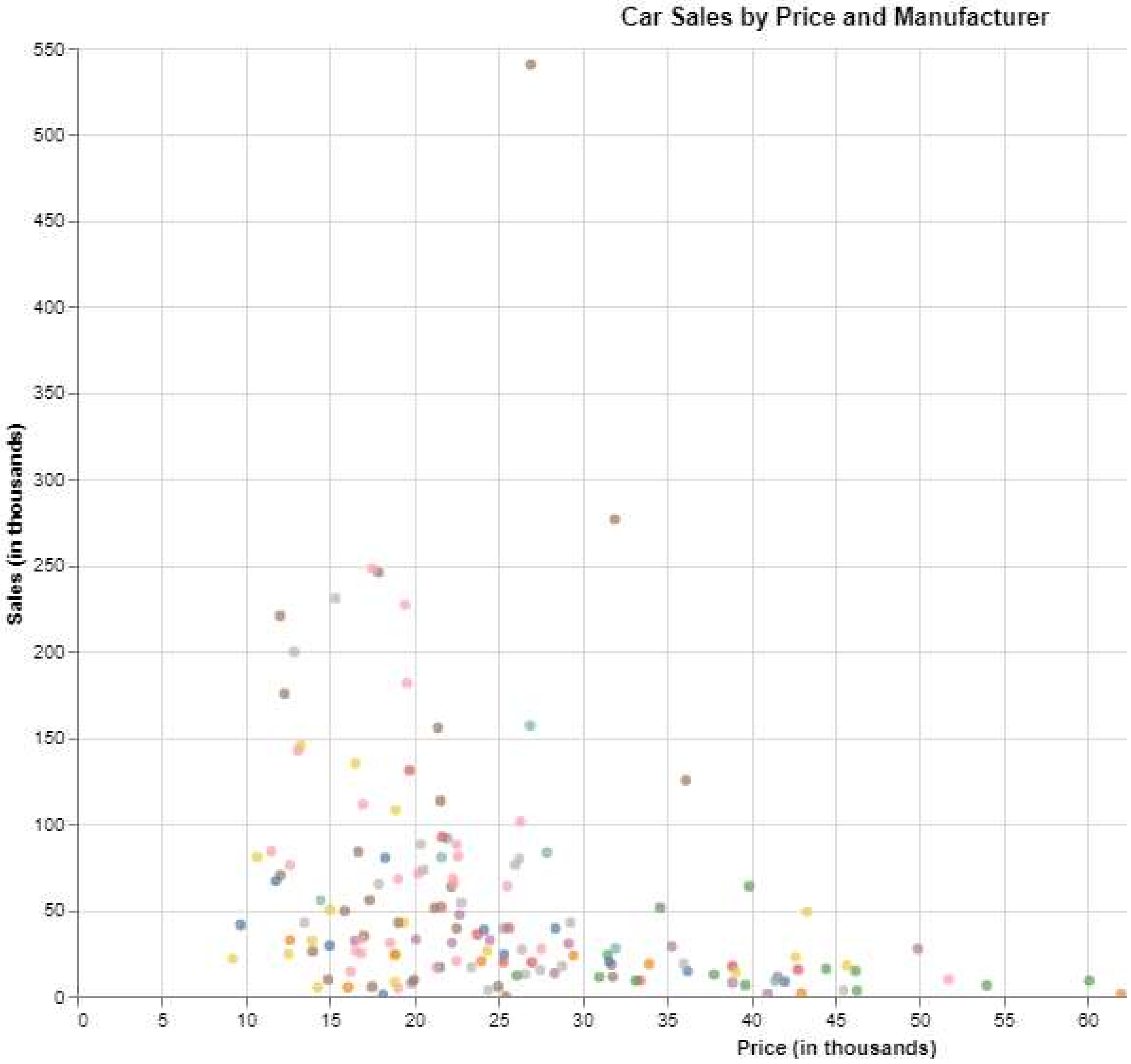
Select_from_the_dropdown_the_car_Manufacturer Acura

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

In [128]:

```
# Create the dropdown selection
dropdown = alt.binding_select(options=list(carsales['Manufacturer'].unique()))
selection = alt.selection_single(fields=['Manufacturer'], bind=dropdown, name=
# 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
```

Out[128]:



Select_from_the_dropdown_the_car_Manufacturer Acura

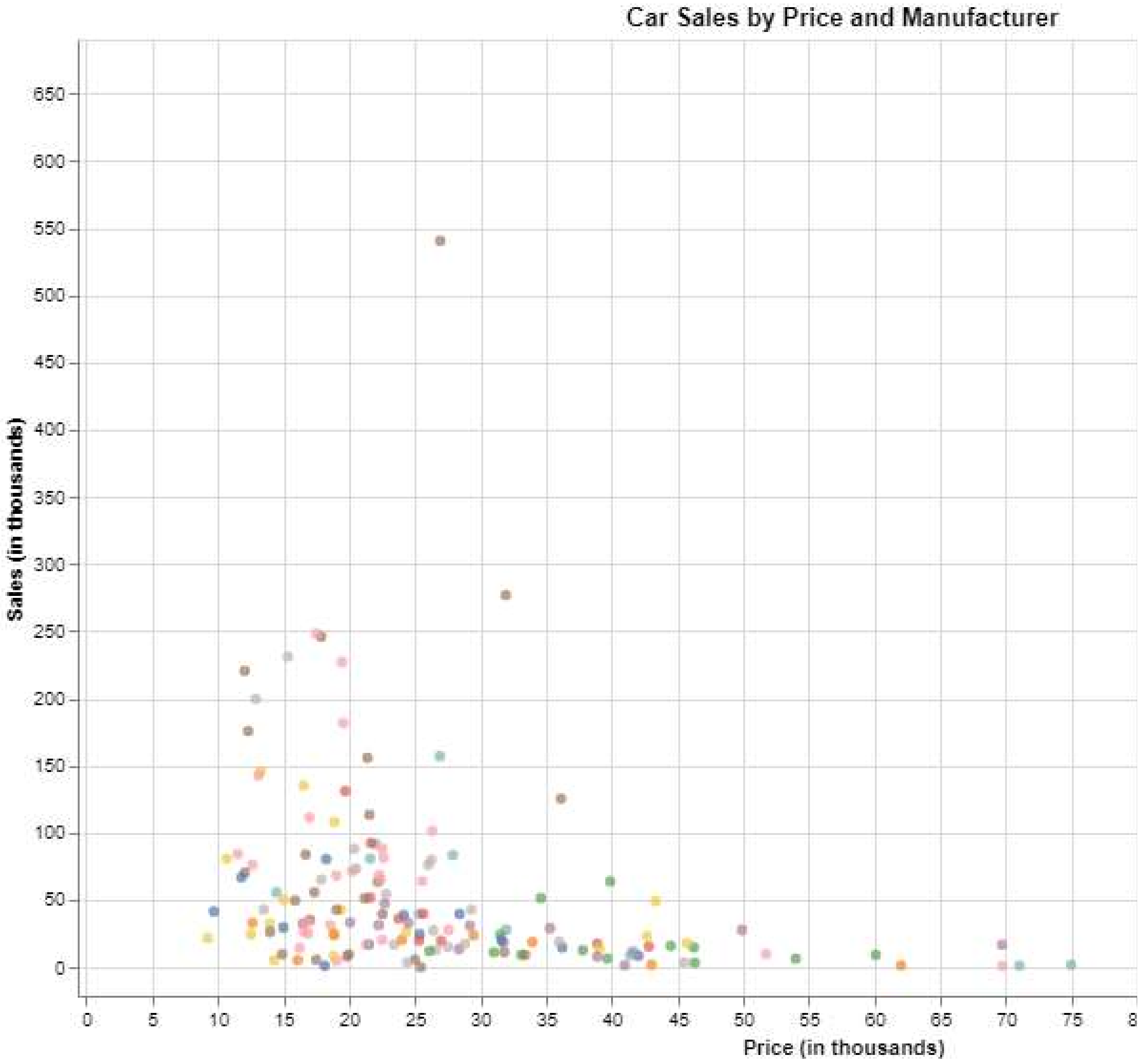
3. ADDING TOOLTIPS

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).

In [129]:

```
# Create the scatter plot
scatterT = 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)',
             alt.Tooltip('Sales_in_thousands:Q', title='Sales (in thousands)',
).add_selection(selection).properties(
    width=800,
    height=500,
    title='Car Sales by Price and Manufacturer'
).interactive()

# Show the plot
scatterT
```

Out[129]:



Select_from_the_dropdown_the_car_Manufacturer Acura

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

In [ ]:

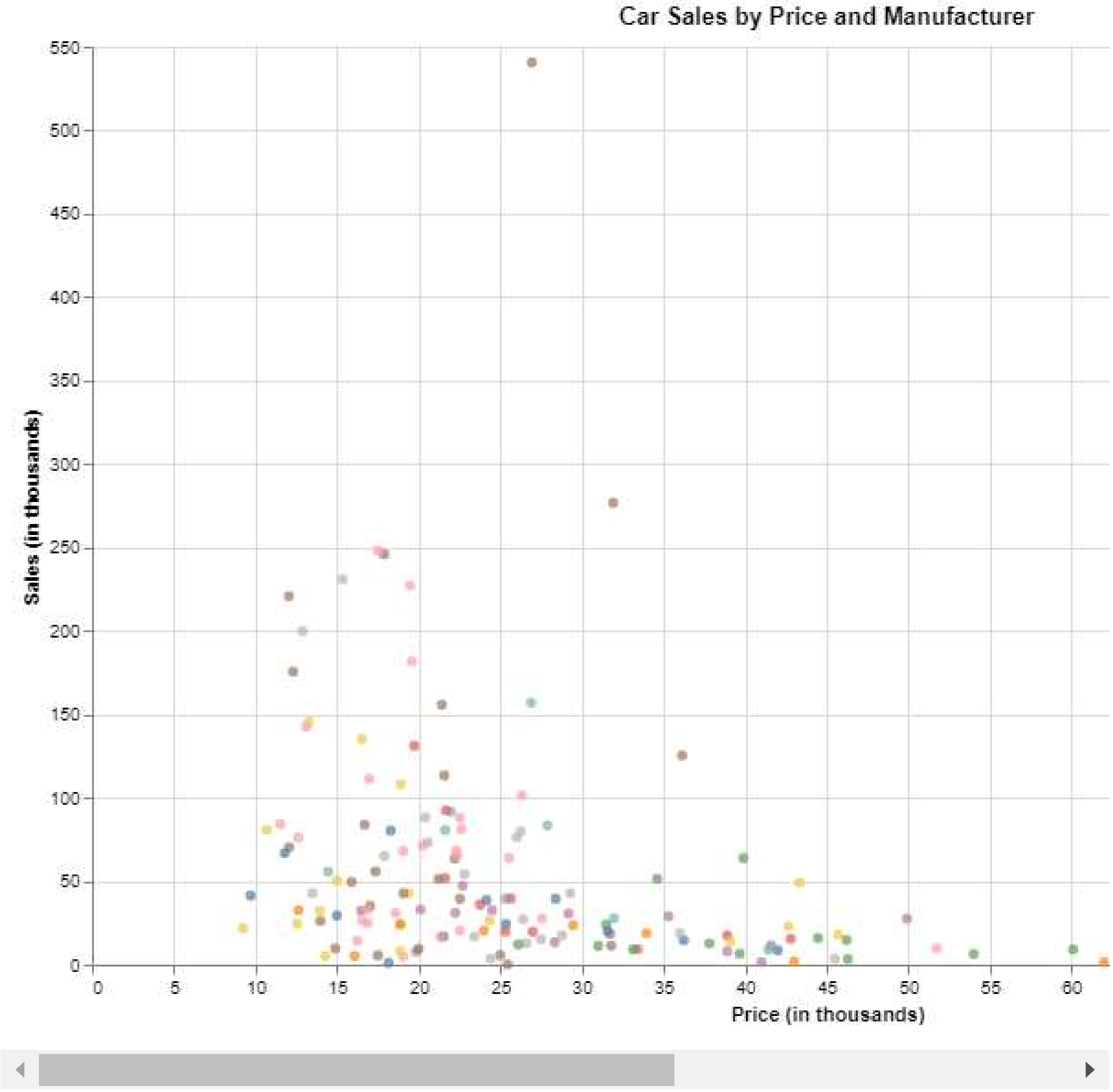
4. SCATTER PLOT WITH HREF

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).

In [130]:

```
# 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_t
).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
```

Out[130]:

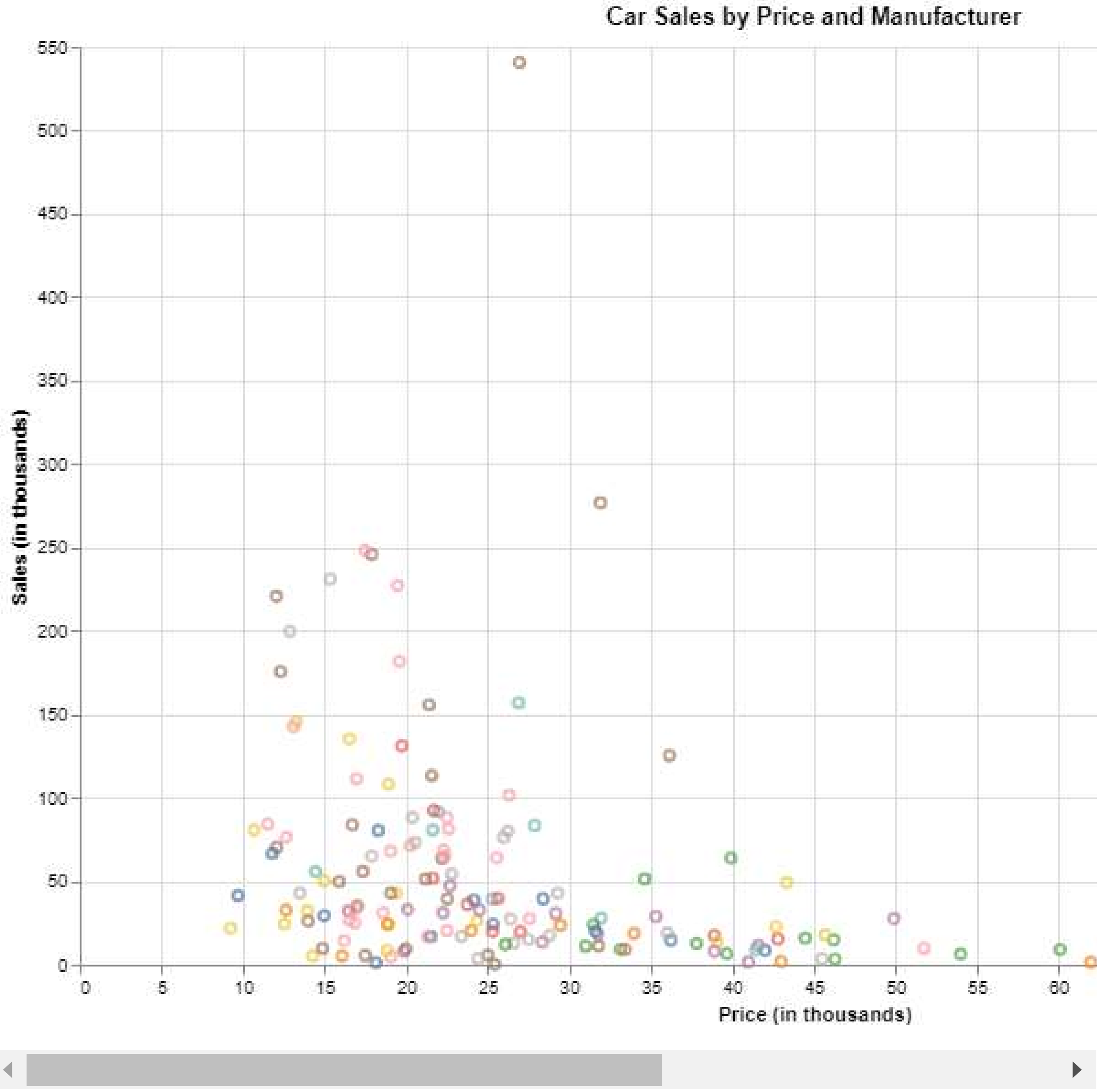

Car Sales by Price and Manufacturer
Sales (in thousands)
550
500
450
400
350
300
250
200
150
100
50
0
0
5
10
15
20
25
30
35
40
45
50
55
60
Price (in thousands)

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.

In [131]:

```
# Create the scatter plot with href
alt.Chart(carsales).transform_calculate(
    url='https://www.google.com/search?q=' + alt.datum.Manufacturer + '+' + al
).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'
)
```

Out[131]:



In [ ]:

In [ ]:

In [ ]:

In [ ]:

In [ ]:

In [ ]: