DATA VISUALIZATION PROJECT:

For the final project, I'm analysing the records from the US Cars dataset.

OBJECTIVE:

The main objective of this project is to analyse the datasets, find unique insights from the data available to us, and visualise using the Altair library.

DATASET:

The dataset is obtained from Kaggle.

LINK: https://www.kaggle.com/code/tanersekmen/us-car-data-analysis-eda-visualization/data

This dataset contains 12 features that describe 2499 vehicles for sale in US.

`Unnamed: 0` - column id.
`price` - price of the car.
`brand` - brand of the car.
`model` - model of the car.
`year` - vehicle registration year.
`title_status` - clean vehicle or salvage insurance.
`mileage` - miles traveled.
`color` - car color.
`vin` - vehicle identification number.
`lot` - A lot number is an identification number assigned to a particular quantity or lot of material from a single.
`state` - The location in which the car is being available for purchase.
`country` - The location in which the car is being available for purchase.
`condition` - time left.

IMPORTING THE DATA:

```
1 import pandas as pd
2 import altair as alt
3 import matplotlib.pyplot as plt
4 from vega_datasets import data
```

```
1 filepath = r"D:\Masters-CU Boulder\Data Visualization\USA_cars_datasets.csv"
```

```
1 df = pd.read_csv(filepath)
2 df
```

	Unnamed: 0	price	brand	model	year	title_status	mileage	color	
0	0	6300	toyota	cruiser	2008	clean vehicle	274117.0	black	j
1	1	2899	ford	se	2011	clean vehicle	190552.0	silver	2fr
2	2	5350	dodge	mpv	2018	clean vehicle	39590.0	silver	3

We drop the first column since it is not required for our analysis.

```
df = df.drop(columns = ['Unnamed: 0'])
```

```
df
```

	price	brand	model	year	title_status	mileage	color	
0	6300	toyota	cruiser	2008	clean vehicle	274117.0	black	jtezu11f88k00
1	2899	ford	se	2011	clean vehicle	190552.0	silver	2fmdk3gc4bbb0
2	5350	dodge	mpv	2018	clean vehicle	39590.0	silver	3c4pdcgg5jt34
3	25000	ford	door	2014	clean vehicle	64146.0	blue	1ftfw1et4efc2
4	27700	chevrolet	1500	2018	clean vehicle	6654.0	red	3gcpcrec2jg47
...	...	...	...	...	...	...	...	
2494	7800	nissan	versa	2019	clean vehicle	23609.0	red	3n1cn7ap9kl88
2495	9200	nissan	versa	2018	clean vehicle	34553.0	silver	3n1cn7ap5jl88
2496	9200	nissan	versa	2018	clean vehicle	31594.0	silver	3n1cn7ap9jl88

CHECKING THE DATASET FOR NULL VALUES:

```
df.isna().sum()
```

```
price           0
brand           0
model           0
year            0
title_status    0
mileage         0
color           0
vin             0
lot             0
state           0
country         0
condition       0
dtype: int64
```

SUMMARY OF THE DATASET:

```
df.dtypes
```

```
price             int64
brand            object
model            object
```

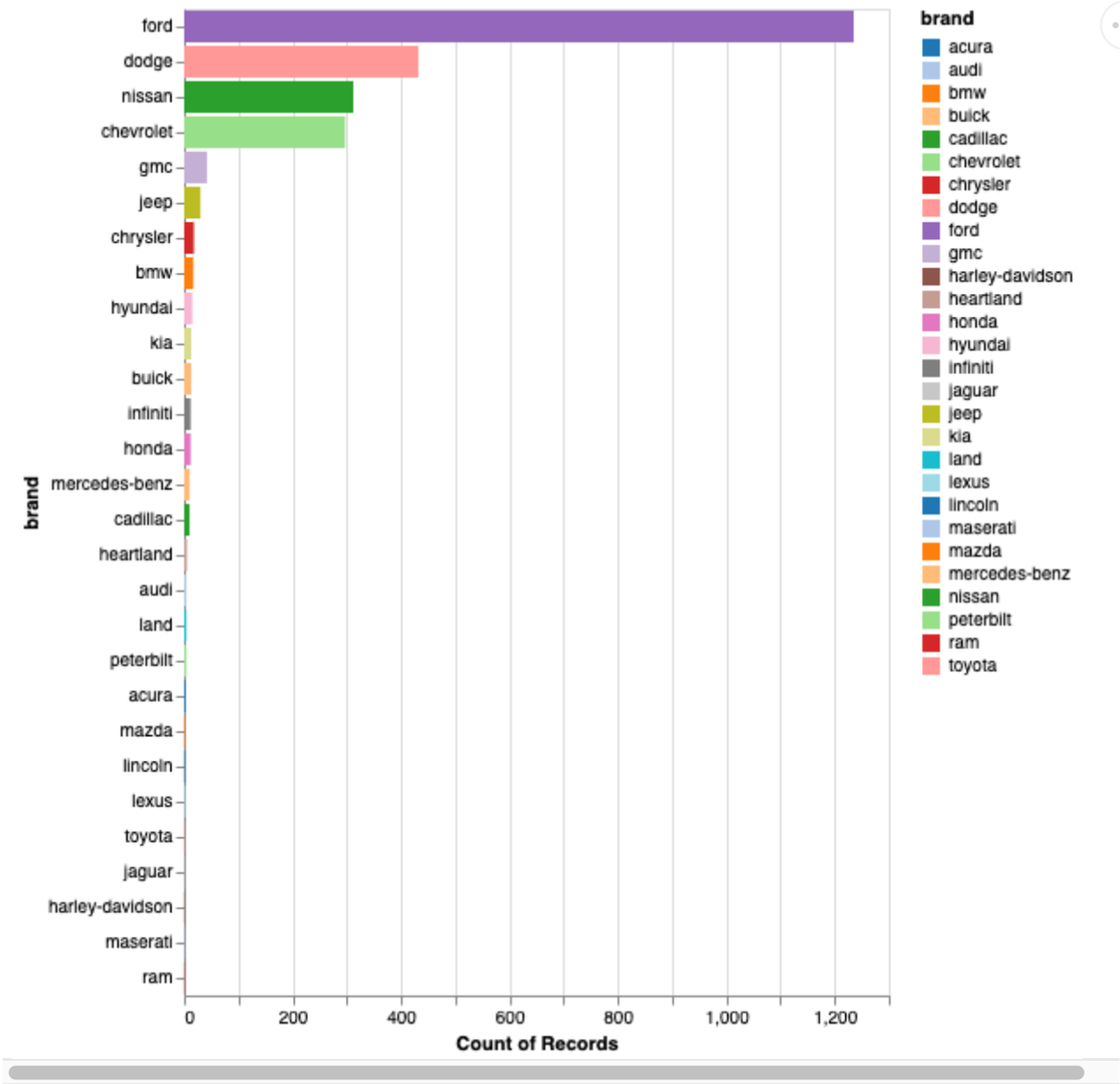
```
year              int64
title_status     object
mileage         float64
color            object
vin              object
lot               int64
state            object
country          object
condition        object
dtype: object
```

IMPLEMENTING VISUALIZATIONS:

VISUALIZATION 1:

The first graph is a simple horizontal barchart plotted using the altair library.
It shows the number of cars available for sale in each brand.

```
alt.Chart(df).mark_bar().encode(
    x='count():Q',
    y=alt.Y('brand', sort='-x'),
    color = alt.Color('brand', scale=alt.Scale(scheme = 'category20'))
)
```

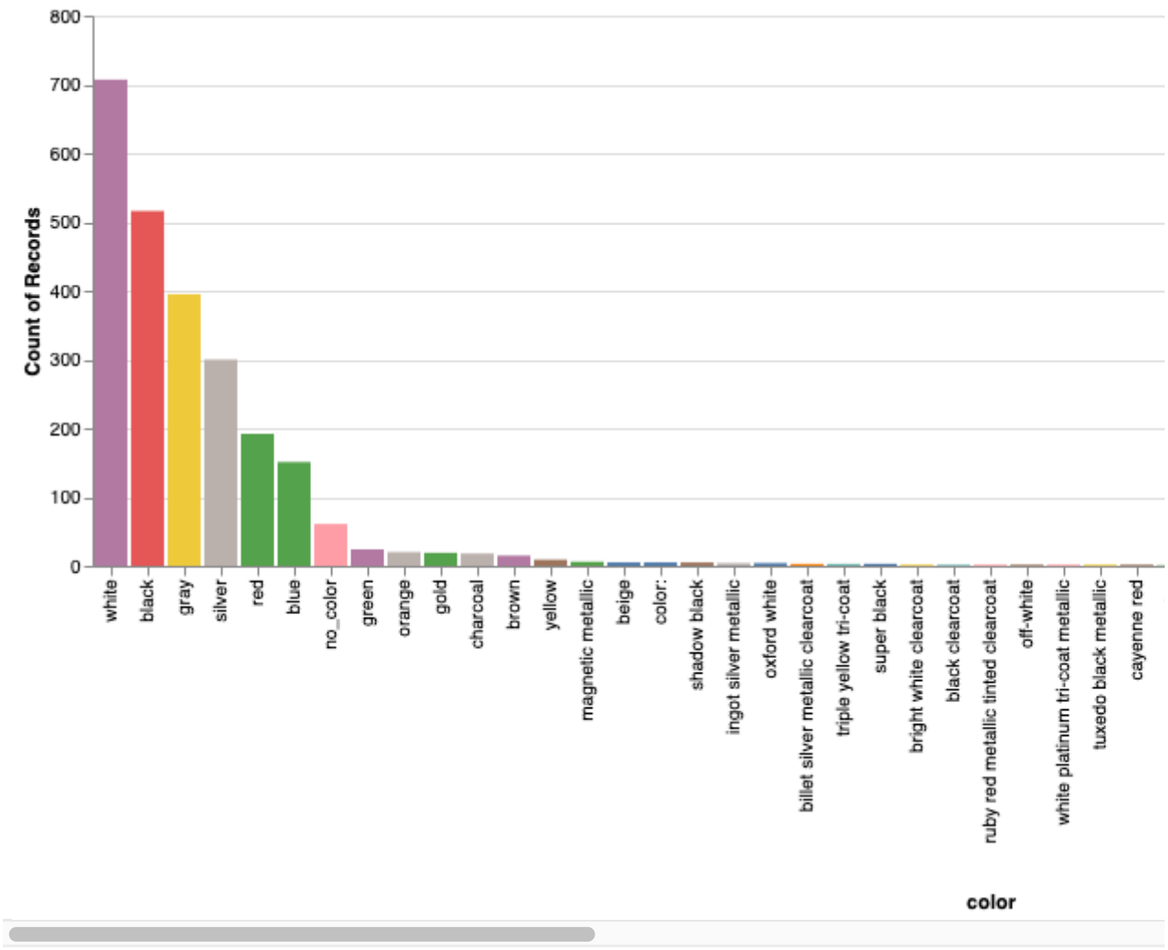


We find out that Ford has the most number of cars for sale, followed by Dodge, Nissan and Chevrolet.

VISUALIZATION 2:

Next plot a vertical barchart to sort the vehicles by color.

```
alt.Chart(df).mark_bar().encode(
    alt.X('color:N', sort='-y'),
    alt.Y('count():Q'),
    color = alt.Color('color', scale=alt.Scale(scheme = 'tableau10')),

)
```



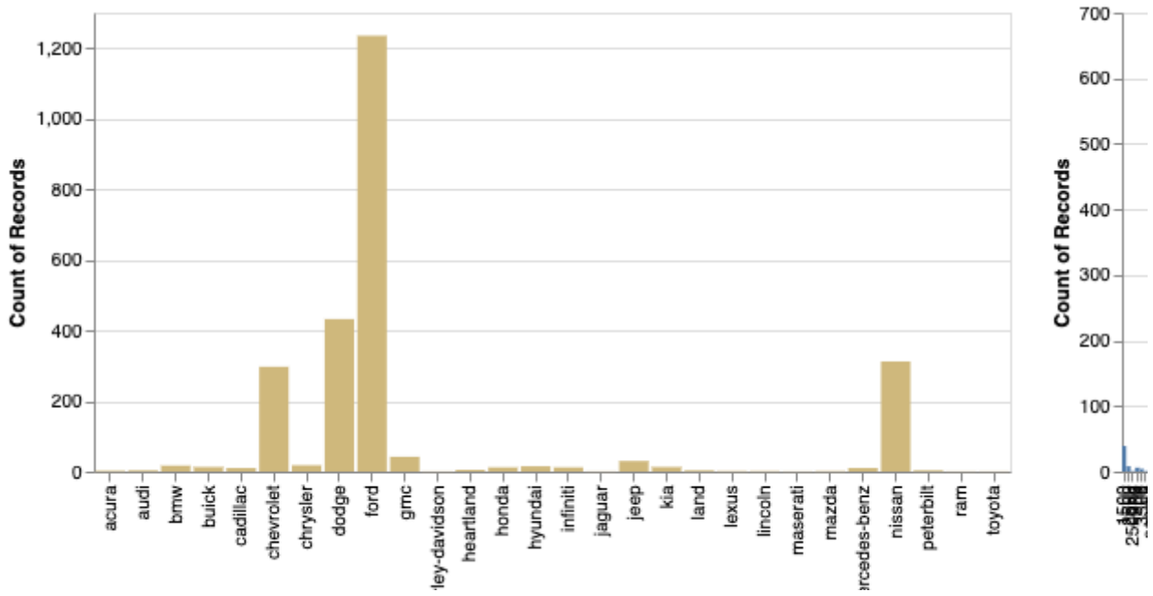
We can see that the most common color is white followed by black, gray, silver, red, blue and so on.

VISUALIZATION 3:

This is an interactive graph in which the overview graph shows the total number cars avilable for sale in each brand.

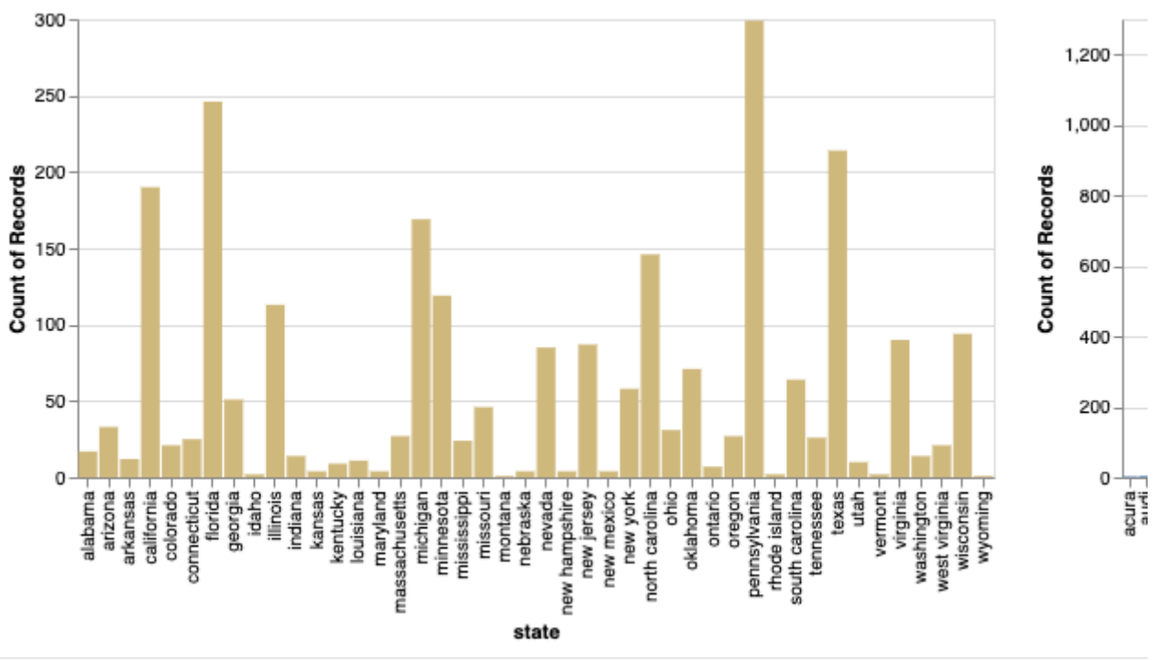
Inthe second graph, which is the details graph, if we click on a particular bar in the first chart, it shows the the number of cars available, by model in the particular brand.

```
selection = alt.selection(type="single", fields=["brand"])

base =  alt.Chart(df).properties(width=500, height=250)

overview = alt.Chart(df).mark_bar().encode(
    x = 'brand',
    y = 'count()',
    color=alt.condition(selection, alt.value("#CFB87C"), alt.value("lightgrey"))
).add_selection(selection).properties(height=250, width=500)

detail = hist = base.mark_bar().encode(
    x = "model",
    y = "count()"
).transform_filter(selection).properties(height=250, width=300)

overview | detail
```



The following chart is similar to the previous chart, but we look at the most popular brands available for sale in each state.

```
selection = alt.selection(type="single", fields=["state"])

base =  alt.Chart(df).properties(width=500, height=250)

overview = alt.Chart(df).mark_bar().encode(
    x = 'state',
    y = 'count()',
    color=alt.condition(selection, alt.value("#CFB87C"), alt.value("lightgrey"))
).add_selection(selection).properties(height=250, width=500)

detail = hist = base.mark_bar().encode(
    x = "brand",
    y = "count()"
).transform_filter(selection).properties(height=250, width=250)

overview | detail
```

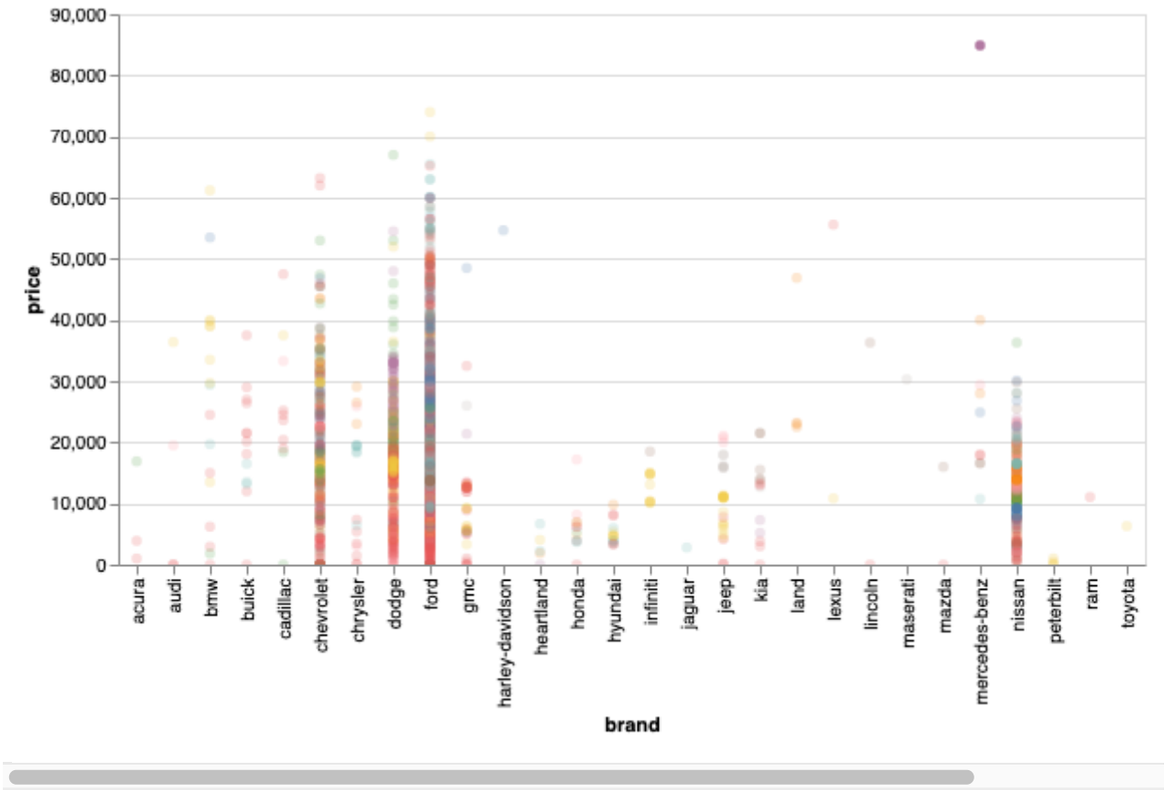


VISUALIZATION 4:

The following interactive scatter plot shows the price of each model by brand. If we hover over a particular point, we can see the details of that model.

```
selection = alt.selection(type='multi', fields=['model'], on='mouseover', nearest=T
```

```
2
3 alt.Chart(df).mark_circle().encode(
4     x = "brand",
5     y = "price",
6     color=alt.Color('model', scale=alt.Scale(scheme='tableau10')),
7     tooltip=["model", "brand", "price"],
8     opacity=alt.condition(selection,alt.value(1),alt.value(.2))
9 ).add_selection(selection).interactive()
```



SUBSETTING THE DATASET:

Most expensive car available in each brand.

```
1 df2 = df.groupby('brand').max()
2 df2.sort_values('price',ascending = False)
```

brand	price	model	year	title_status	mileage	color	
mercedes-benz	84900	vans	2019	clean vehicle	110907.0	white	wddzf4jb6ha2
ford	74000	wagon	2020	salvage insurance	999999.0	yellow	wf0dp3th0g41
dodge	67000	van	2020	salvage insurance	239822.0	white	3d7ks28c15g7
chevrolet	63200	volt	2020	salvage insurance	507985.0	yellow	kl8cb6sa3jc4
bmw	61200	x3	2020	salvage insurance	216657.0	white	wbawb33548p1
lexus	55600	mpv	2020	clean vehicle	36596.0	silver	jtjam7bx4l52
harley-davidson	54680	road/street	2016	clean vehicle	9502.0	black	1hd1krm1xgb6
gmc	48500	mpv	2019	salvage insurance	235348.0	white	2gtv2mecxk11

```
df.year.unique()
```

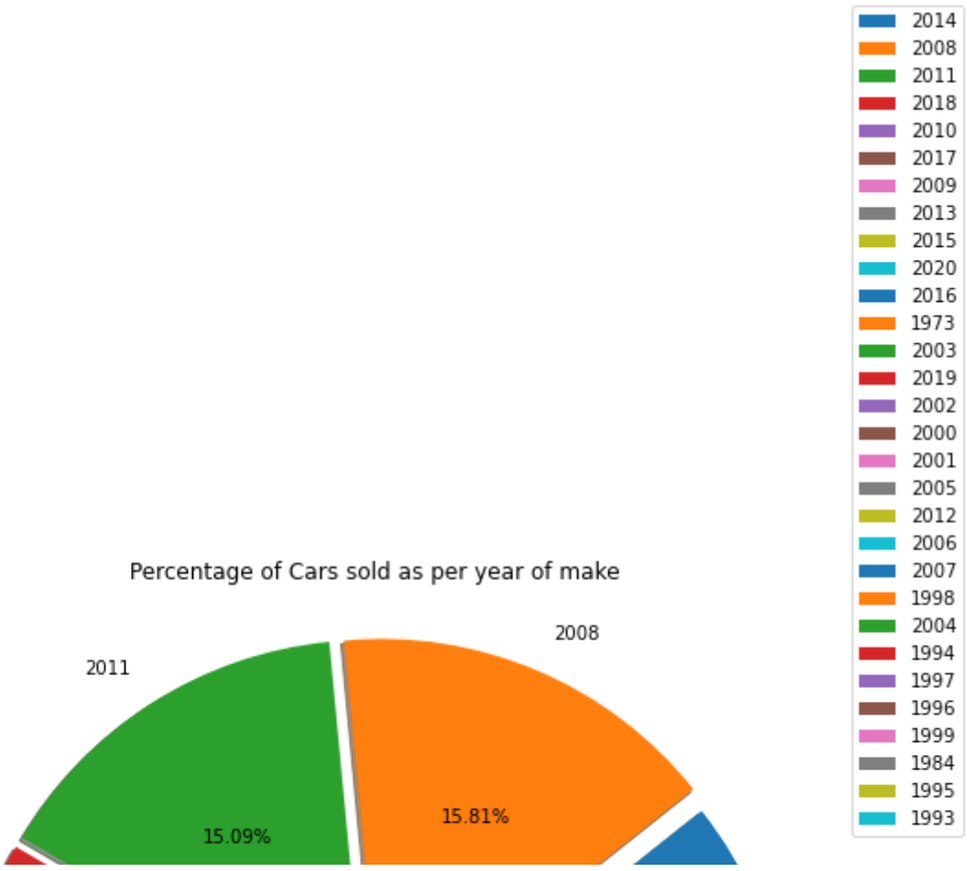
```
array([2008, 2011, 2018, 2014, 2010, 2017, 2009, 2013, 2015, 2020, 2016,
       1973, 2003, 2019, 2002, 2000, 2001, 2005, 2012, 2006, 2007, 1998,
       2004, 1994, 1997, 1996, 1999, 1984, 1995, 1993], dtype=int64)
```

insurance

VISUALIZATION 5:

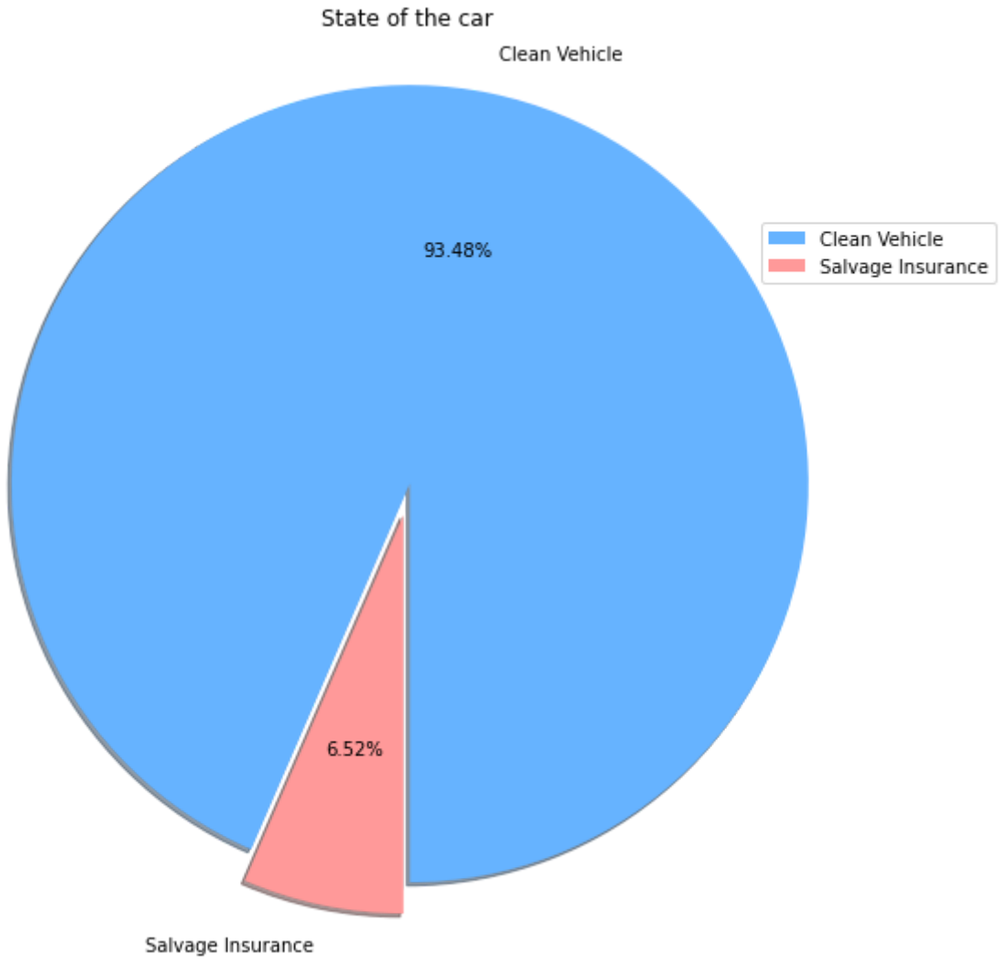
We plot the percentage of cars by the year they were registered in a pie chart. Since Pie-Charts are not available in altair, we use matplotlib instead.

```
s = df['year'].value_counts()
label = [r'2014', r'2008', r'2011', r'2018', r'2010', r'2017', r'2009', r'2013', r'
          r'1973', r'2003', r'2019', r'2002', r'2000', r'2001', r'2005', r'2012',
          r'2004', r'1994', r'1997', r'1996', r'1999', r'1984', r'1995', r'1993']

plt.pie(s, labels = label, shadow=True,explode=(0.1, 0.1, 0.1,0.1, 0.1, 0.1,0.1, 0.
plt.title('Percentage of Cars sold as per year of make',  pad = 150)
plt.legend(bbox_to_anchor=(1.0, 1.02, 1., 1.202), loc=4)
plt.show()
```



The following Pie chart shows the status of the vehicles on sale.

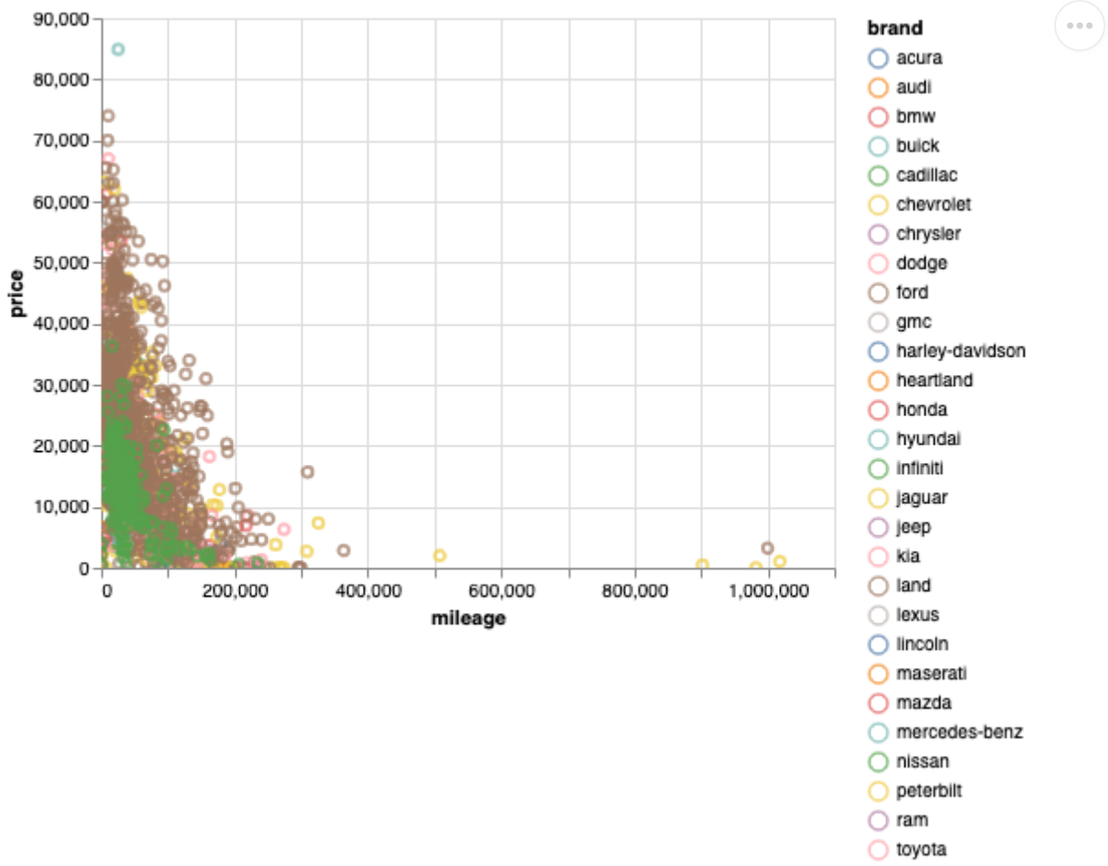
```
s1 = df['title_status'].value_counts()
label = [r'Clean Vehicle', r'Salvage Insurance']

plt.pie(s1, labels = label, shadow=True, explode=(0.1, 0.1), colors=['#66b3ff', '#f
plt.title('Status of the car',  pad = 150)
plt.legend(bbox_to_anchor=(1.0, 1.02, 1., 1.202), loc=4)
plt.show()
```



VISUALIZATION 6:

We plot this interactive scatter plot to check if there is a relationship between the mileage and the price of the car.

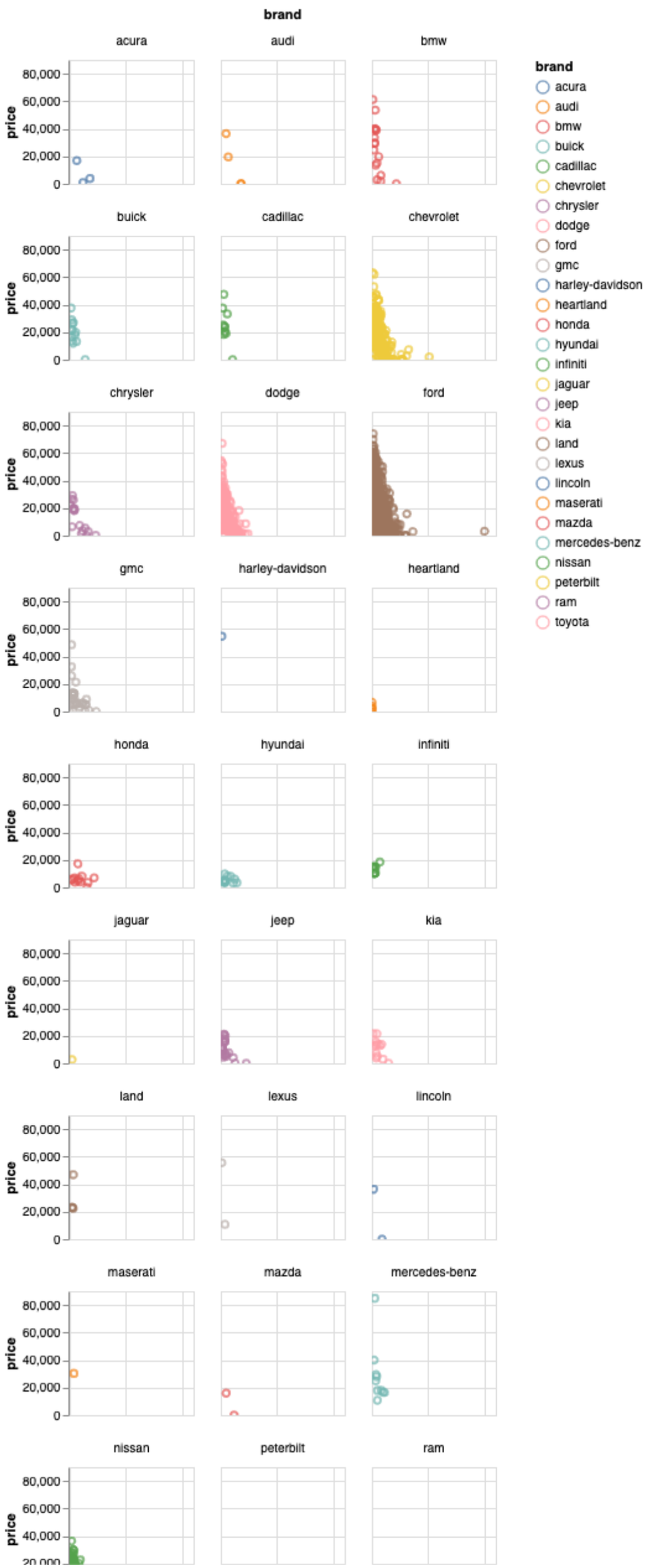
```
alt.Chart(df).mark_point().encode(
    x='mileage:Q',
    y='price:Q',
    color='brand:N',
    tooltip=['brand', 'model', 'mileage', 'year', 'price']
).interactive()
```



We can see that there is an exponential relationship. The prices of the cars drop significantly when the mileage increases.

We further confirm this by looking at their relationship by brand in the faceted scatterplot below.

```
alt.Chart(df).mark_point().encode(
    x = alt.X('mileage'),
    y = alt.Y('price'),
    color = 'brand'
).properties(width = 100, height = 100).facet(
    'brand:N',
    columns = 3
).interactive()
```


brand
acura
audi
bmw
buick
cadillac
chevrolet
chrysler
dodge
ford
gmc
harley-davidson
heartland
honda
hyundai
infiniti
jaguar
jeep
kia
land
lexus
lincoln
maserati
mazda
mercedes-benz
nissan
peterbilt
ram
price
80,000
60,000
40,000
20,000
0
brand
acura
audi
bmw
buick
cadillac
chevrolet
chrysler
dodge
ford
gmc
harley-davidson
heartland
honda
hyundai
infiniti
jaguar
jeep
kia
land
lexus
lincoln
maserati
mazda
mercedes-benz
nissan
peterbilt
ram
toyota

Hence, we can confirm the same relationship we found previously.

DESIGN ELEMENTS:

LIBRARIES USED:

1. altair.
2. matplotlib.
3. vega_datasets.

DESIGN ELEMENTS USED:

1. Bar Chart.
2. Interactive bar chart using altair.
3. Pie Chart.
4. Scatter Plot.

Since we are looking to find new insights from the dataset, the usage of the design elements mentioned above were best suited to achieve our objectives.

1. Bar charts are great at displaying the count of categorical data.
2. Making the bar charts interactive allows us to show the data represented by each bar in detail and in a user-friendly manner.
3. While pie charts are usually not preferred, they represent percentage values in a simple and effective manner.
4. The scatter plot was used to plot the details of the car by brand and also to plot the relationship between mileage and price.

EVALUATION:

I've had 3 people evaluate the project.

1. My parents.
2. Friend.
3. Classmate.

Their opinion was to do away with **VISUALIZATION 4** that had an interactive scatter-plot showing the price and other details of a particular car, because they were a little difficult to interpret. But I decided to keep it since it showed multiple relationships in a single chart.

FINDINGS:

1. The top 3 brands based on availability:
    - Ford.
    - Dodge.
    - Nissan.

1. The top 3 colors based on availability"
    - White.
    - Black.
    - Gray.

3. The costliest car available is a 2019 Mercedes-Benz Vans priced at $84900.

4. 35.69% of the cars on sale were registered in 2014, 15.81% in 2008, 15.09 in 2011, 8.12% in 2018, 7.84% in 2010 and less than 5% of the cars were registered in each of the remaining years.

5. 93.48% percentage of the cars are clean vehicles while the remaining 6.52% are salvaged.

6. There is an exponential relationship between mileage and price of the car across all brands. The price of the car drops significantly when mileage increases.

1