```python
In [2]: import pandas as pd
```

```python
In [3]: import numpy as np
```

```python
In [4]: import plotly.express as px
```

```python
In [5]: data =pd.read_csv("apple_products (1).csv")
```

```python
In [5]: data
```

Out[5]:

	Product Name	Product URL	Brand	Sale Price	Mrp	Discount Percentage	Number Of Ratings	Number Of Reviews	
0	APPLE iPhone 8 Plus (Gold, 64 GB)	https://www.flipkart.com/apple-iphone-8-plus-g...	Apple	49900	49900	0	3431	356	MOBEXRGV7EHI
1	APPLE iPhone 8 Plus (Space Grey, 256 GB)	https://www.flipkart.com/apple-iphone-8-plus-s...	Apple	84900	84900	0	3431	356	MOBEXRGVAC
2	APPLE iPhone 8 Plus (Silver, 256 GB)	https://www.flipkart.com/apple-iphone-8-plus-s...	Apple	84900	84900	0	3431	356	MOBEXRGVGET/
3	APPLE iPhone 8 (Silver, 256 GB)	https://www.flipkart.com/apple-iphone-8-silver...	Apple	77000	77000	0	11202	794	MOBEXRGVMZW
4	APPLE iPhone 8 (Gold, 256 GB)	https://www.flipkart.com/apple-iphone-8-gold-2...	Apple	77000	77000	0	11202	794	MOBEXRGVPK7
...	...	...	...	...	...	...	...	...	
57	APPLE iPhone SE (Black, 64 GB)	https://www.flipkart.com/apple-iphone-se-black...	Apple	29999	39900	24	95909	8161	MOBFWQ6BR3M
58	APPLE iPhone 11 (Purple, 64 GB)	https://www.flipkart.com/apple-iphone-11-purpl...	Apple	46999	54900	14	43470	3331	MOBFWQ6BTFF
59	APPLE iPhone 11 (White, 64 GB)	https://www.flipkart.com/apple-iphone-11-white...	Apple	46999	54900	14	43470	3331	MOBFWQ6BVWV
60	APPLE iPhone 11 (Black, 64 GB)	https://www.flipkart.com/apple-iphone-11-black...	Apple	46999	54900	14	43470	3331	MOBFWQ6BXGJ
61	APPLE iPhone 11 (Red, 64 GB)	https://www.flipkart.com/apple-iphone-11-red-6...	Apple	46999	54900	14	43470	3331	MOBFWQ6BYYV

62 rows × 11 columns

In [6]:
```python
print(data.isnull().sum())
```

Loading [MathJax]/extensions/Safe.js

```
Product Name          0
Product URL           0
Brand                 0
Sale Price            0
Mrp                   0
Discount Percentage   0
Number Of Ratings     0
Number Of Reviews     0
Upc                   0
Star Rating           0
Ram                   0
dtype: int64
```

In [7]: `print(data.describe())`

```
            Sale Price             Mrp   Discount Percentage   Number Of Ratings  \
count        62.000000      62.000000             62.000000           62.000000
mean      80073.887097   88058.064516              9.951613        22420.403226
std       34310.446132   34728.825597              7.608079        33768.589550
min       29999.000000   39900.000000              0.000000          542.000000
25%       49900.000000   54900.000000              6.000000          740.000000
50%       75900.000000   79900.000000             10.000000         2101.000000
75%      117100.000000  120950.000000             14.000000        43470.000000
max      140900.000000  149900.000000             29.000000        95909.000000

         Number Of Reviews   Star Rating
count            62.000000     62.000000
mean           1861.677419      4.575806
std            2855.883830      0.059190
min              42.000000      4.500000
25%              64.000000      4.500000
50%             180.000000      4.600000
75%            3331.000000      4.600000
max            8161.000000      4.700000
```

iphone sales analysis

In [8]:
```python
highest_rated = data.sort_values(by=["Star Rating"],ascending = False)
highest_rated =highest_rated.head(10)
print(highest_rated["Product Name"])
```
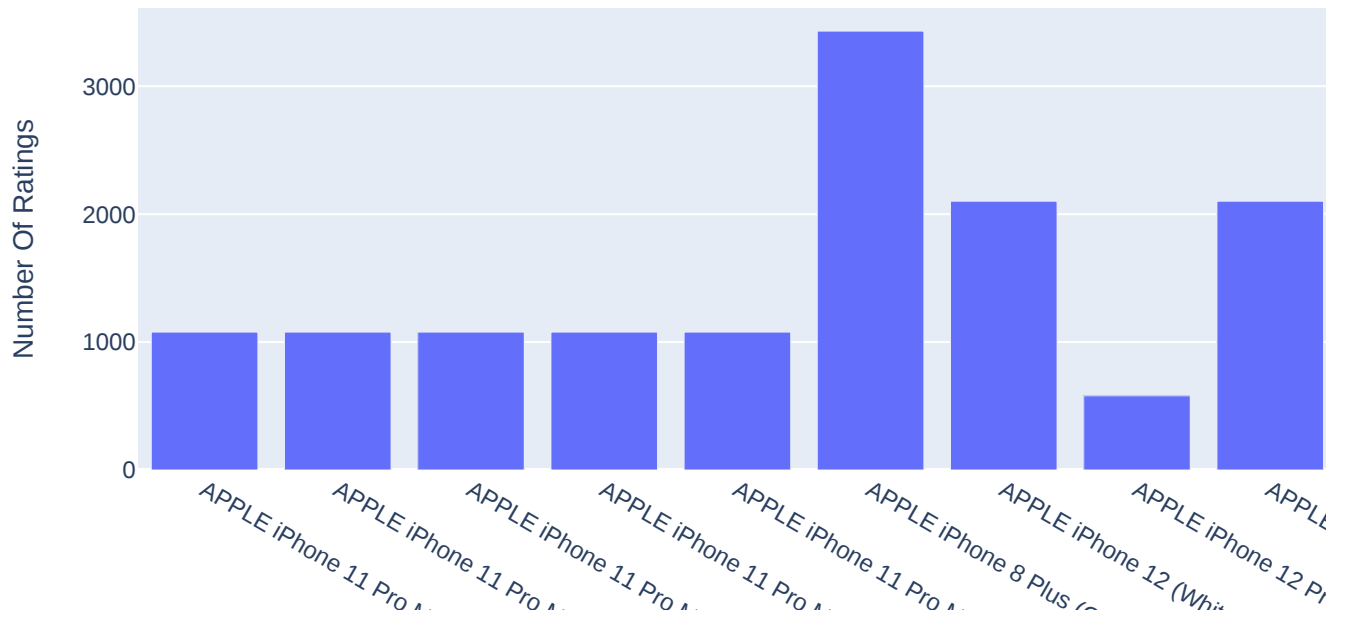
```
20       APPLE iPhone 11 Pro Max (Midnight Green, 64 GB)
17           APPLE iPhone 11 Pro Max (Space Grey, 64 GB)
16      APPLE iPhone 11 Pro Max (Midnight Green, 256 GB)
15               APPLE iPhone 11 Pro Max (Gold, 64 GB)
14              APPLE iPhone 11 Pro Max (Gold, 256 GB)
0                     APPLE iPhone 8 Plus (Gold, 64 GB)
29                    APPLE iPhone 12 (White, 128 GB)
32           APPLE iPhone 12 Pro Max (Graphite, 128 GB)
35                    APPLE iPhone 12 (Black, 128 GB)
36                    APPLE iPhone 12 (Blue, 128 GB)
Name: Product Name, dtype: object
```

In [9]:
```python
iphones= highest_rated["Product Name"].value_counts()
lables=iphones.index
counts=highest_rated["Number Of Ratings"]
figure= px.bar(highest_rated,x=lables,y=counts,
            title="Number of Ratings of Highest Rated Phone")
figure.show()
```
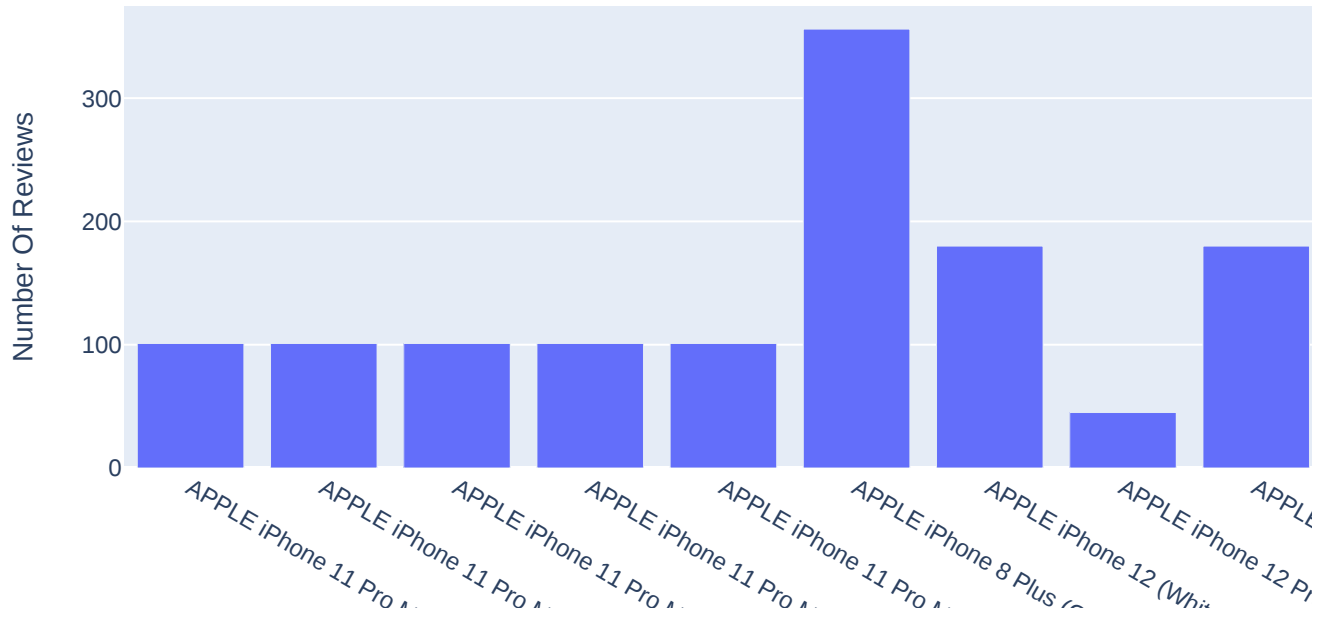
Loading [MathJax]/extensions/Safe.js

Number of Ratings of Highest Rated Phone


```
In [10]:    iphones= highest_rated["Product Name"].value_counts()
            lables=iphones.index
            counts=highest_rated["Number Of Reviews"]
            figure= px.bar(highest_rated,x=lables,y=counts,
                        title="Number of Reviews of Highest Rated Phone")
            figure.show()
```
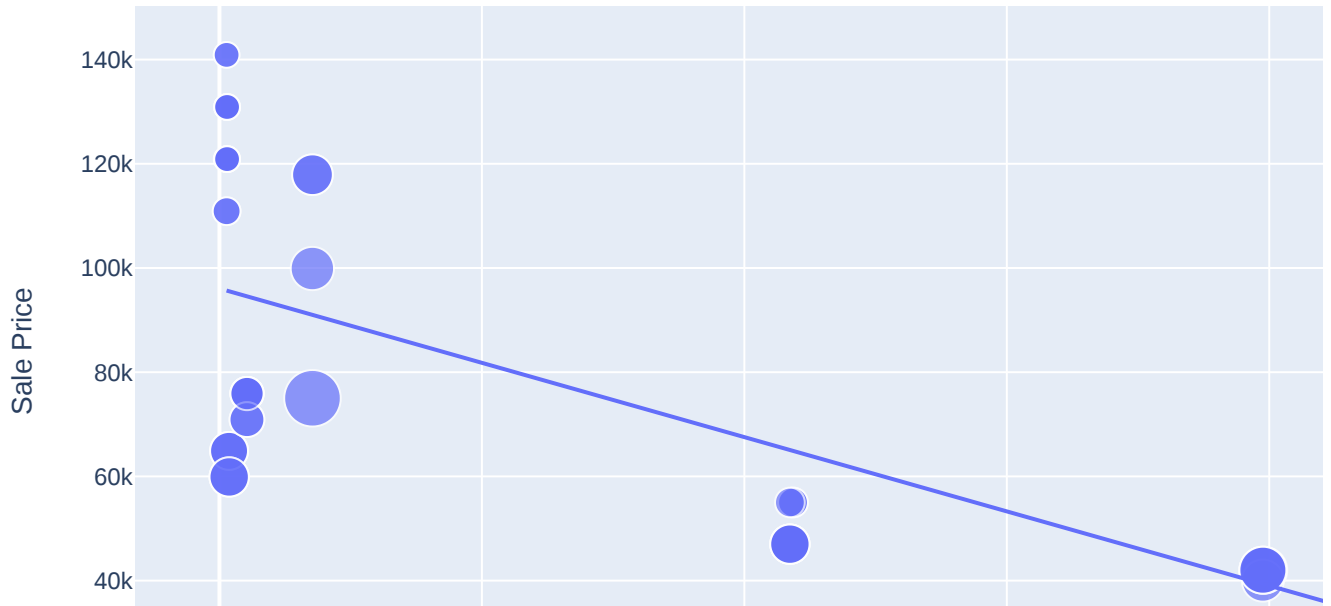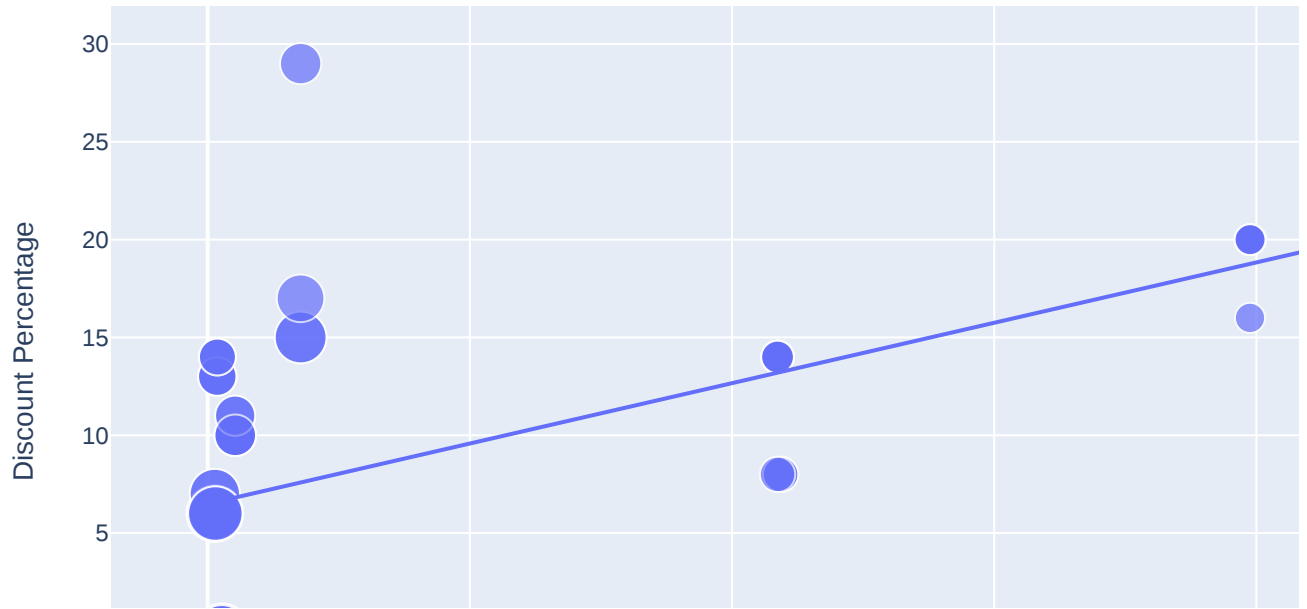
Number of Reviews of Highest Rated Phone


```
In [11]:   figure =px.scatter(data_frame=data,x="Number Of Ratings",
                              y="Sale Price",size= "Discount Percentage",
                              trendline ="ols",
                              title="Relationship between sales price and number of ratings")
           figure.show()
```

Relationship between sales price and number of ratings


```
In [12]:   figure=px.scatter(data_frame=data,
                            x="Number Of Ratings",
                           y="Discount Percentage",
                           size="Sale Price",
                           trendline="ols",
                           title="Relationship between discount percentage number of ratings")
           figure.show()
```

Relationship between discount percentage number of ratings

In [ ]:

In [ ]: