This notebook contains the Data Exploration Analysis for Sprocket Plt Company.

I will be answering their inquires about their marketing strategy, using the insights from this dataset.

After every charts the deductions follows to explain the results

In [1]:
```
#Importing relevant packages
import numpy as np
import pandas as pd
import seaborn as sns
import matplotlib.pyplot as plt

%matplotlib inline

```

Extracting the data file from my device

In [2]:
```
df = pd.read_excel('KPMG_customer_data.xlsx', sheet_name=None)

# the sheets names 'Title Sheet',NewClients'Transactions', 'NewCustomerList', 'Custo
```

```
C:\Users\MRS COLLINS\AppData\Local\Temp\ipykernel_6216\169425300.py:1: FutureWarning: I
nferring datetime64[ns] from data containing strings is deprecated and will be removed
in a future version. To retain the old behavior explicitly pass Series(data, dtype=date
time64[ns])
  df = pd.read_excel('KPMG_customer_data.xlsx', sheet_name=None)
C:\Users\MRS COLLINS\AppData\Local\Temp\ipykernel_6216\169425300.py:1: FutureWarning: I
nferring datetime64[ns] from data containing strings is deprecated and will be removed
in a future version. To retain the old behavior explicitly pass Series(data, dtype=date
time64[ns])
  df = pd.read_excel('KPMG_customer_data.xlsx', sheet_name=None)
C:\Users\MRS COLLINS\AppData\Local\Temp\ipykernel_6216\169425300.py:1: FutureWarning: I
nferring datetime64[ns] from data containing strings is deprecated and will be removed
in a future version. To retain the old behavior explicitly pass Series(data, dtype=date
time64[ns])
  df = pd.read_excel('KPMG_customer_data.xlsx', sheet_name=None)
```

The above warining is as regards to one of the sheet that contains a date type as number, we'll ingnore it because we'er not working on that sheet

The 1000 new customer's data information

understanding the dataset

In [3]:

```
clients1 = df['NewClients']
clients1
```

Out[3]:

	first_name	last_name	gender	past_3_years_bike_related_purchases	DOB	job_title	job_indus
0	Chickie	Brister	Male	86	1957-07-12	General Manager	I
1	Morly	Genery	Male	69	1970-03-22	Structural Engineer	
2	Ardelis	Forrester	Female	10	1974-08-28	Senior Cost Accountant	Fina
3	Lucine	Stutt	Female	64	1979-01-28	Account Representative III	I
4	Melinda	Hadlee	Female	34	1965-09-21	Financial Analyst	Fina
...	...	...	...	...	...	...	
995	Ferdinand	Romanetti	Male	60	1959-10-07	Paralegal	Fina
996	Burk	Wortley	Male	22	2001-10-17	Senior Sales Associate	
997	Melloney	Temby	Female	17	1954-10-05	Budget/Accounting Analyst IV	Fina
998	Dickie	Cubbini	Male	30	1952-12-17	Financial Advisor	Fina
999	Sylas	Duffill	Male	56	1955-10-02	Staff Accountant IV	

1000 rows × 23 columns

In [4]:
```
clients1.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 1000 entries, 0 to 999
Data columns (total 23 columns):
 #   Column                               Non-Null Count  Dtype
---  ------                               --------------  -----
 0   first_name                           1000 non-null   object
 1   last_name                            971 non-null    object
 2   gender                               1000 non-null   object
 3   past_3_years_bike_related_purchases  1000 non-null   int64
 4   DOB                                  983 non-null    datetime64[ns]
 5   job_title                            894 non-null    object
 6   job_industry_category                835 non-null    object
 7   wealth_segment                       1000 non-null   object
 8   deceased_indicator                   1000 non-null   object
 9   owns_car                             1000 non-null   object
 10  tenure                               1000 non-null   int64
 11  address                              1000 non-null   object
 12  postcode                             1000 non-null   int64
 13  state                                1000 non-null   object
 14  country                              1000 non-null   object
 15  property_valuation                   1000 non-null   int64
 16  Unnamed: 16                          1000 non-null   float64
 17  Unnamed: 17                          1000 non-null   float64
 18  Unnamed: 18                          1000 non-null   float64
 19  Unnamed: 19                          1000 non-null   float64
 20  Unnamed: 20                          1000 non-null   int64
 21  Rank                                 1000 non-null   int64
 22  Value                                1000 non-null   float64
dtypes: datetime64[ns](1), float64(5), int64(6), object(11)
memory usage: 179.8+ KB
```

In [5]:
```
clients1.describe()
```

Out[5]:

	past_3_years_bike_related_purchases	tenure	postcode	property_valuation	Unnamed: 16	Un
count	1000.000000	1000.000000	1000.000000	1000.000000	1000.000000	1000
mean	49.836000	11.388000	3019.227000	7.397000	0.751790	0
std	27.796686	5.037145	848.895767	2.758804	0.206927	0
min	0.000000	0.000000	2000.000000	1.000000	0.400000	0
25%	26.750000	7.000000	2209.000000	6.000000	0.580000	0
50%	51.000000	11.000000	2800.000000	8.000000	0.740000	0
75%	72.000000	15.000000	3845.500000	9.000000	0.930000	1
max	99.000000	22.000000	4879.000000	12.000000	1.100000	1

In [ ]:

To begin data cleaning, we check for data types and missing values

In [6]:
```
clients1.dtypes
```

Out[6]:
```
first_name                                   object
last_name                                    object
gender                                       object
past_3_years_bike_related_purchases           int64
DOB                                  datetime64[ns]
job_title                                    object
job_industry_category                        object
wealth_segment                               object
deceased_indicator                           object
owns_car                                     object
tenure                                        int64
address                                      object
postcode                                      int64
state                                        object
country                                      object
property_valuation                            int64
Unnamed: 16                                 float64
Unnamed: 17                                 float64
Unnamed: 18                                 float64
Unnamed: 19                                 float64
Unnamed: 20                                   int64
Rank                                          int64
Value                                       float64
dtype: object
```

In [7]:
```
clients1.isna().sum()
```

Out[7]:
```
first_name                               0
last_name                               29
gender                                   0
past_3_years_bike_related_purchases      0
DOB                                     17
job_title                              106
job_industry_category                  165
wealth_segment                           0
deceased_indicator                       0
owns_car                                 0
tenure                                   0
address                                  0
postcode                                 0
state                                    0
country                                  0
property_valuation                       0
Unnamed: 16                              0
Unnamed: 17                              0
Unnamed: 18                              0
Unnamed: 19                              0
Unnamed: 20                              0
Rank                                     0
Value                                    0
dtype: int64
```

There are missing values in columns: last_name, DOB, job_title and job_industry_category

For the purpose of this analysis, I will drop thee first_name, last_name, decreased_indicator, Address, postcode,country, rank and value.

Reasons:

- first_name, last_name, postcode and adrress are personal information
- the decreased indicator is a genaral N
- contry has nor relevance because they aare all autralia
- Rank and value rates from 1 ro 1000, they don't infer with the analysis

NOTE:

The column names Unnamed's are irrelevant to this analysis, those information are viod.

In [8]:
```
# Dropping some of the columns
n_clients1 = clients1.drop(['first_name', 'last_name', 'address', 'postcode', 'dece
                            'Unnamed: 16','Unnamed: 17', 'Unnamed: 18', 'Unnamed: 1
                            'Unnamed: 20', 'country', 'Rank', 'Value'], axis = 1)
```

In [9]:
```
# remane the past_3_year
n_clients1.rename(columns={'past_3_years_bike_related_purchases': 'bike_purchase'},
```

In [10]:
```
n_clients1.head()
```

Out[10]:

	gender	bike_purchase	DOB	job_title	job_industry_category	wealth_segment	owns_car	tenure	s
0	Male	86	1957-07-12	General Manager	Manufacturing	Mass Customer	Yes	14	(
1	Male	69	1970-03-22	Structural Engineer	Property	Mass Customer	No	16	N
2	Female	10	1974-08-28	Senior Cost Accountant	Financial Services	Affluent Customer	No	10	
3	Female	64	1979-01-28	Account Representative III	Manufacturing	Affluent Customer	Yes	5	(
4	Female	34	1965-09-21	Financial Analyst	Financial Services	Affluent Customer	No	19	N

Explore the values in each columns to get an idea of their inputs

In [11]:
```
n_clients1['wealth_segment'].unique()
```

Out[11]:
```
array(['Mass Customer', 'Affluent Customer', 'High Net Worth'],
      dtype=object)
```

In [12]:
```
n_clients1['gender'].unique()
```
Out[12]:
```
array(['Male', 'Female', 'U'], dtype=object)
```

In [44]:
```
n_clients1['owns_car'].unique()
```
Out[44]:
```
array(['Yes', 'No'], dtype=object)
```

In [45]:
```
n_clients1['state'].unique()
```
Out[45]:
```
array(['QLD', 'NSW', 'VIC'], dtype=object)
```

In [14]:
```
n_clients1['job_industry_category'].unique()
```
Out[14]:
```
array(['Manufacturing', 'Property', 'Financial Services', 'Entertainment',
       'Retail', 'IT', 'Telecommunications', 'Health', nan, 'Argiculture'],
      dtype=object)
```

remove nulls and missplelt entries

In [15]:
```
# from gender remove 'u'

u = n_clients1[(n_clients1['gender'] == 'U')].index
u
```
Out[15]:
```
Int64Index([ 59, 226, 324, 358, 360, 374, 434, 439, 574, 598, 664, 751, 775,
            835, 883, 904, 984],
           dtype='int64')
```

In [16]:
```
n_clients1.drop(u, inplace=True)
```

In [17]:
```
n_clients1['gender'].unique()
```
Out[17]:
```
array(['Male', 'Female'], dtype=object)
```

In [18]:
```
# drop nulls in DOB, job_title and job_industry_category

clean_set = n_clients1.dropna(axis = 0, how = 'any')
```

In [19]:
```
# the cleaned datsets
clean_set.isna().sum()
```
Out[19]:
```
gender                   0
bike_purchase            0
DOB                      0
job_title                0
job_industry_category    0
wealth_segment           0
owns_car                 0
tenure                   0
state                    0
property_valuation       0
dtype: int64
```

Checking each data types

In [20]:
```
# The data types of the cleaned dataset
clean_set.dtypes
```

Out[20]:
```
gender                            object
bike_purchase                      int64
DOB                       datetime64[ns]
job_title                         object
job_industry_category             object
wealth_segment                    object
owns_car                          object
tenure                             int64
state                             object
property_valuation                 int64
dtype: object
```

In [21]:
```
# The data we are working with conatins 735 rows(observations) and 10 columns(Feilds
clean_set
```

Out[21]:

	gender	bike_purchase	DOB	job_title	job_industry_category	wealth_segment	owns_car	tenu
0	Male	86	1957-07-12	General Manager	Manufacturing	Mass Customer	Yes	
1	Male	69	1970-03-22	Structural Engineer	Property	Mass Customer	No	
2	Female	10	1974-08-28	Senior Cost Accountant	Financial Services	Affluent Customer	No	
3	Female	64	1979-01-28	Account Representative III	Manufacturing	Affluent Customer	Yes	
4	Female	34	1965-09-21	Financial Analyst	Financial Services	Affluent Customer	No	
...	...	...	...	...	...	...	...	
995	Male	60	1959-10-07	Paralegal	Financial Services	Affluent Customer	No	
996	Male	22	2001-10-17	Senior Sales Associate	Health	Mass Customer	No	
997	Female	17	1954-10-05	Budget/Accounting Analyst IV	Financial Services	Affluent Customer	Yes	
998	Male	30	1952-12-17	Financial Advisor	Financial Services	Mass Customer	Yes	
999	Male	56	1955-10-02	Staff Accountant IV	Property	Mass Customer	Yes	

735 rows × 10 columns

In [ ]:
```

```

count relevant columns

In [22]:
```
wealth = clean_set['wealth_segment'].value_counts()
wealth
```

Out[22]:
```
Mass Customer        369
High Net Worth       184
Affluent Customer    182
Name: wealth_segment, dtype: int64
```

In [23]:
```
states = clean_set['state'].value_counts()
states
```

Out[23]:
```
NSW    364
VIC    200
QLD    171
Name: state, dtype: int64
```

In [24]:
```
car_owners= clean_set['owns_car'].value_counts()
car_owners
```

Out[24]:
```
No     376
Yes    359
Name: owns_car, dtype: int64
```

In [25]:
```
job_industry = clean_set['job_industry_category'].value_counts()
job_industry
```

Out[25]:
```
Financial Services    187
Manufacturing         175
Health                138
Retail                 73
Property               51
Entertainment          34
IT                     30
Argiculture            24
Telecommunications     23
Name: job_industry_category, dtype: int64
```

In [26]:
```
gender = clean_set['gender'].value_counts()
gender
```

Out[26]:
```
Female    380
Male      355
Name: gender, dtype: int64
```

Data Exploration Analysis and Visualization on the new customer data

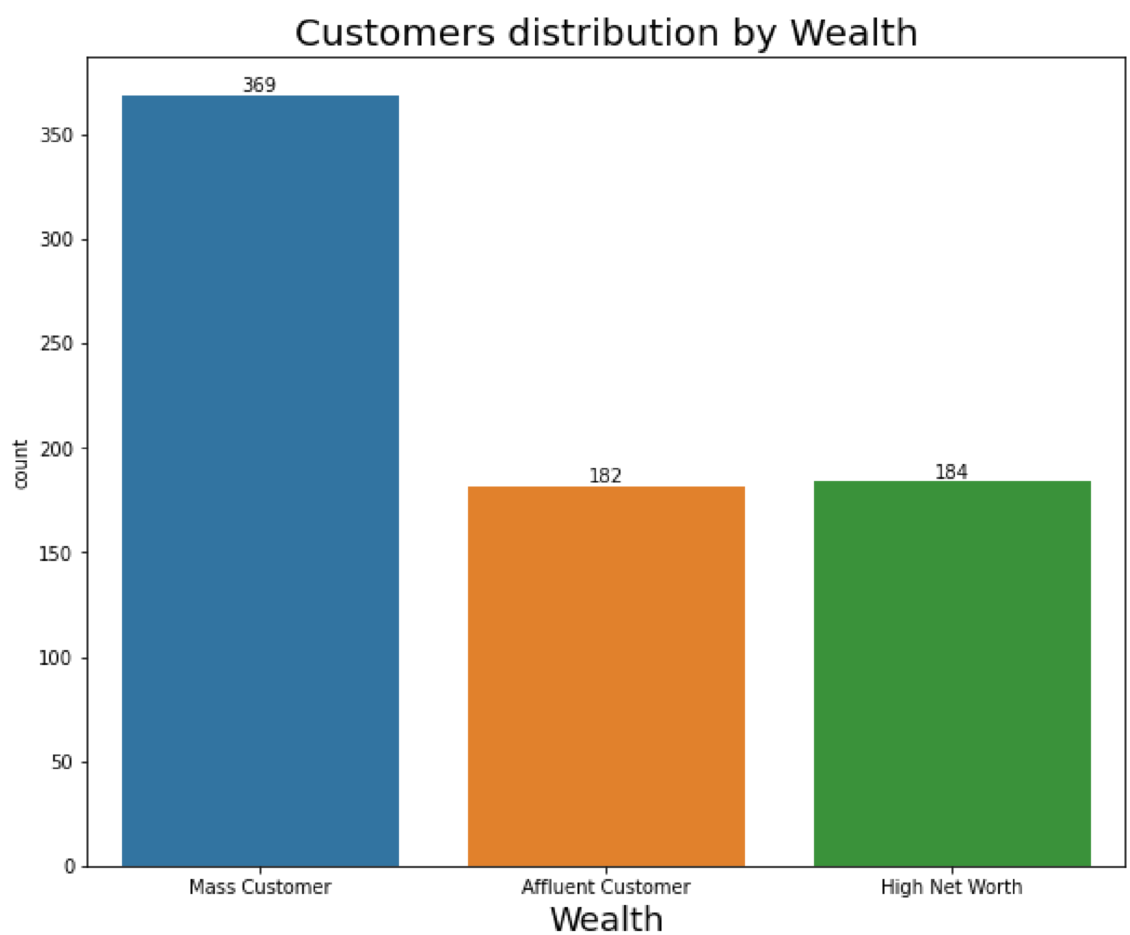
Which customer segment has the highest customer value?

Segments of interest:

- Wealth
- Gender

- Job industry
- State

In [48]:
```
# To know which segment has the highest customer values by wealth

fig, ax = plt.subplots(figsize=(10, 8))


plot= sns.countplot(x='wealth_segment', data=clean_set, ax =ax)
#sns.color_palette("husl", 4)
ax.set_title('Customers distribution by Wealth', fontsize=20)
ax.set_xlabel('Wealth', fontsize =18)
ax.bar_label(ax.containers[0])
plt.show()
```

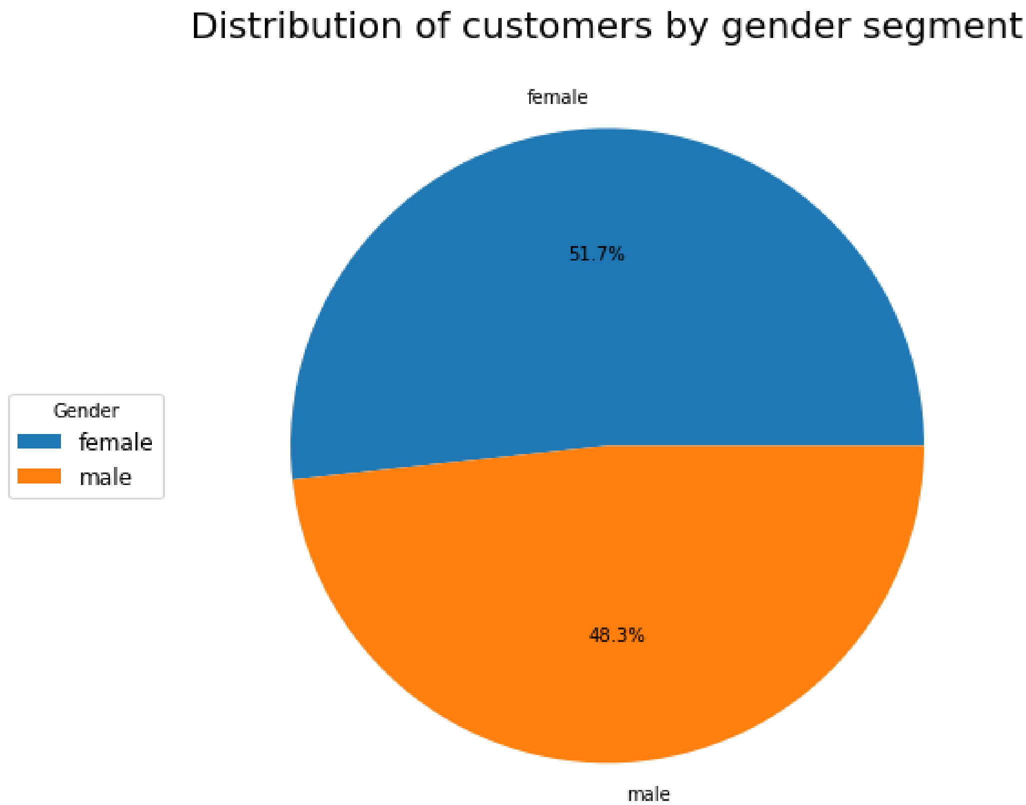


Deductions:

From the charts we can infer that the "mass customer" segment has the highest customer value

- The marketting team of Sprocket Plt should focus their campaigns towards the Mass Customers
- Equal chances should be given to their Affluent and High Net Worth customers

In [49]:
```
# To know which segment has the highest customer values by gender

fig = plt.figure(figsize=(12,8))
plt.axis('equal')
plt.pie(gender, labels = ['female', 'male'], autopct='%1.1f%%')
plt.title('Distribution of customers by gender segment', fontsize =20)
plt.legend(loc = 'center left', title = 'Gender', fontsize = 12)
plt.show()
```



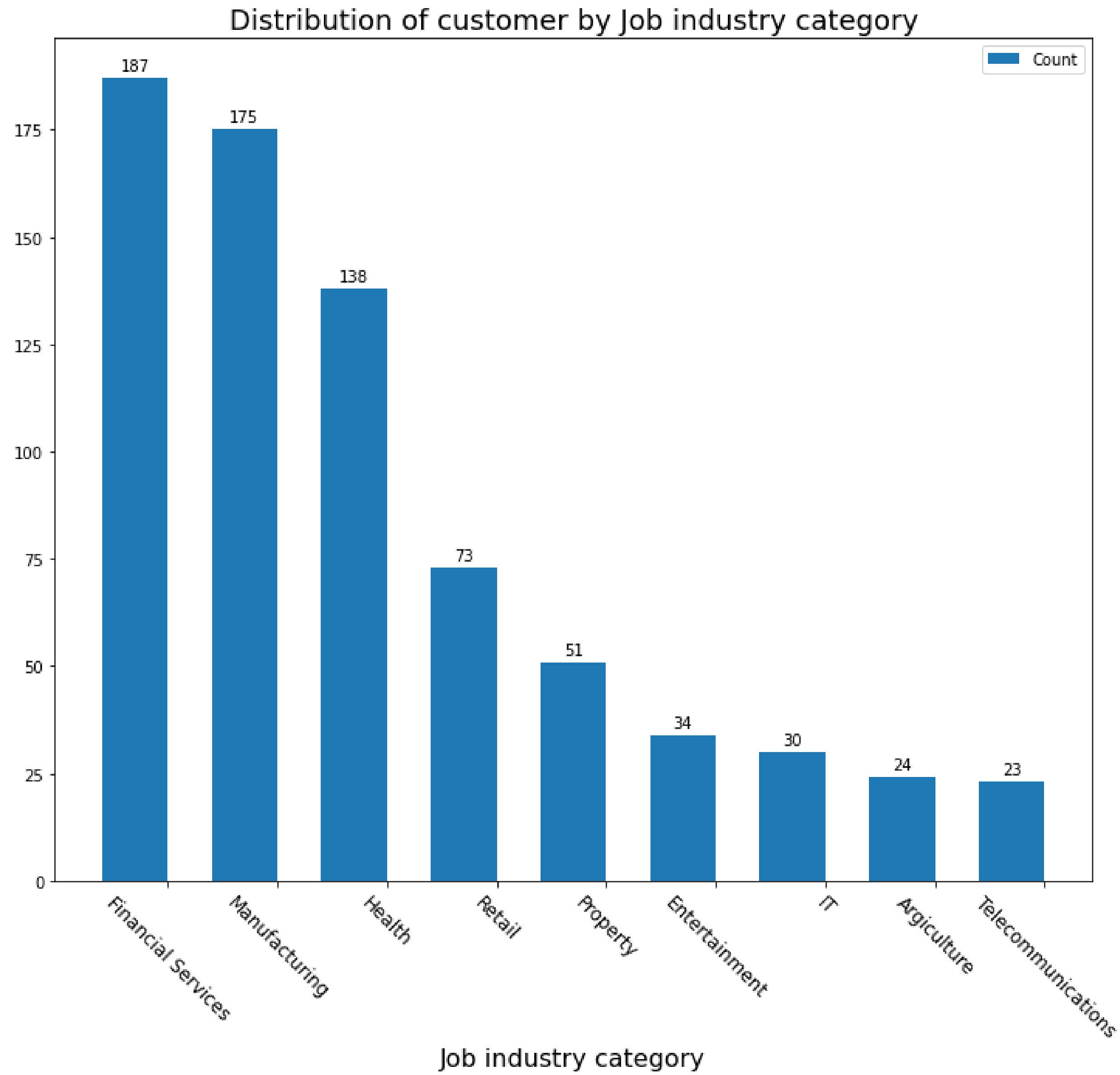
Deductions:

The illustration above tells us that the Female segment has 51.7% of customer value.

- The marketting team should launch their campaign towards both gender while considering the female first
- Since the difference between male and female isn't that much, both customers have value as regards to Sprocket plt.

In [47]:

```
# To know which segment has the highest customer values by Job industry category

industry = ['Financial Services', 'Manufacturing','Health', 'Retail', 'Property','E
            'IT','Argiculture','Telecommunications' ]
count= [187,175,138 , 73, 51,34, 30,24, 23 ]  # These industry and count was gotten


x = np.arange(len(job_industry))
width = .6 #the width of the bar
fig, ax = plt.subplots(figsize=(12, 10))
bar1 = ax.bar(x - width/2, job_industry, width, label='Count')


ax.set_title('Distribution of customer by Job industry category', fontsize=18)
ax.set_xlabel('Job industry category', fontsize=16)
ax.set_xticks(x, industry, rotation=-45, fontsize=12)
ax.legend()

ax.bar_label(bar1, padding=3,)



plt.show()
```



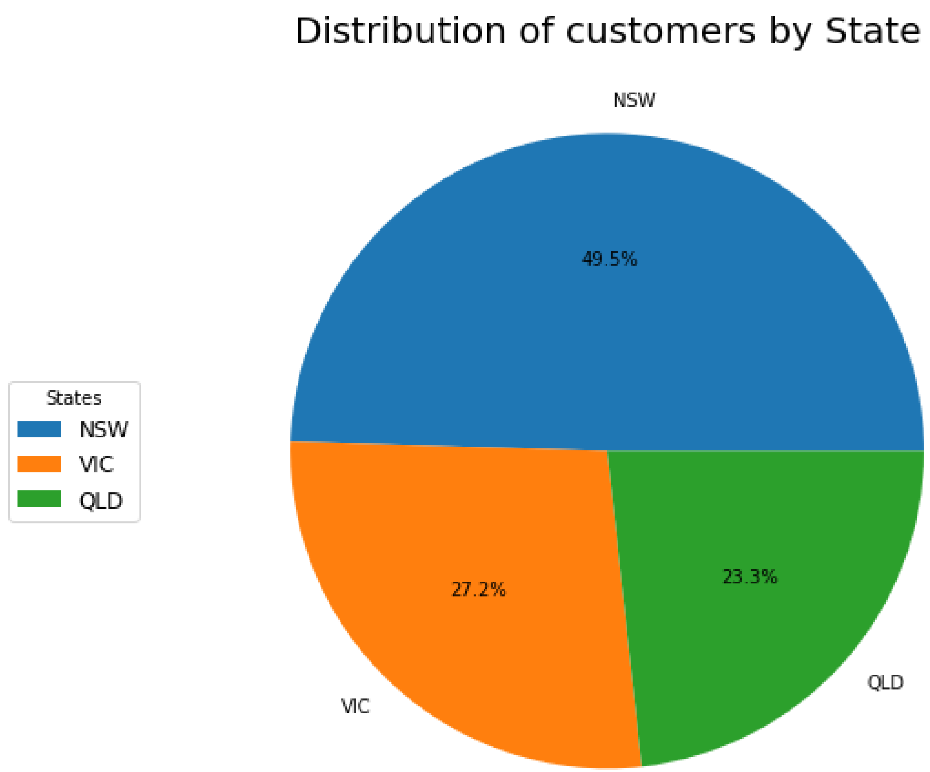
Deductions:

The bar chart reveals that the industry segment with thw highest customer value is, Financial Services(187)

- The marketting team of Spocket plt should leverage their customers that work in the "Finanacial services, Manufacturing, and Health sector".
- Close attention should be paid to the financial and manufacturing indurstry categories

In [50]:

```
# To know which segment has the highest customer values by States

fig = plt.figure(figsize=(12,8))
plt.axis('equal')
plt.pie(states, labels = ['NSW', 'VIC', 'QLD'], autopct='%1.1f%%', )
plt.title('Distribution of customers by State', fontsize =20)
plt.legend(loc = 'center left', title = 'States', fontsize = 12)
plt.show()
```

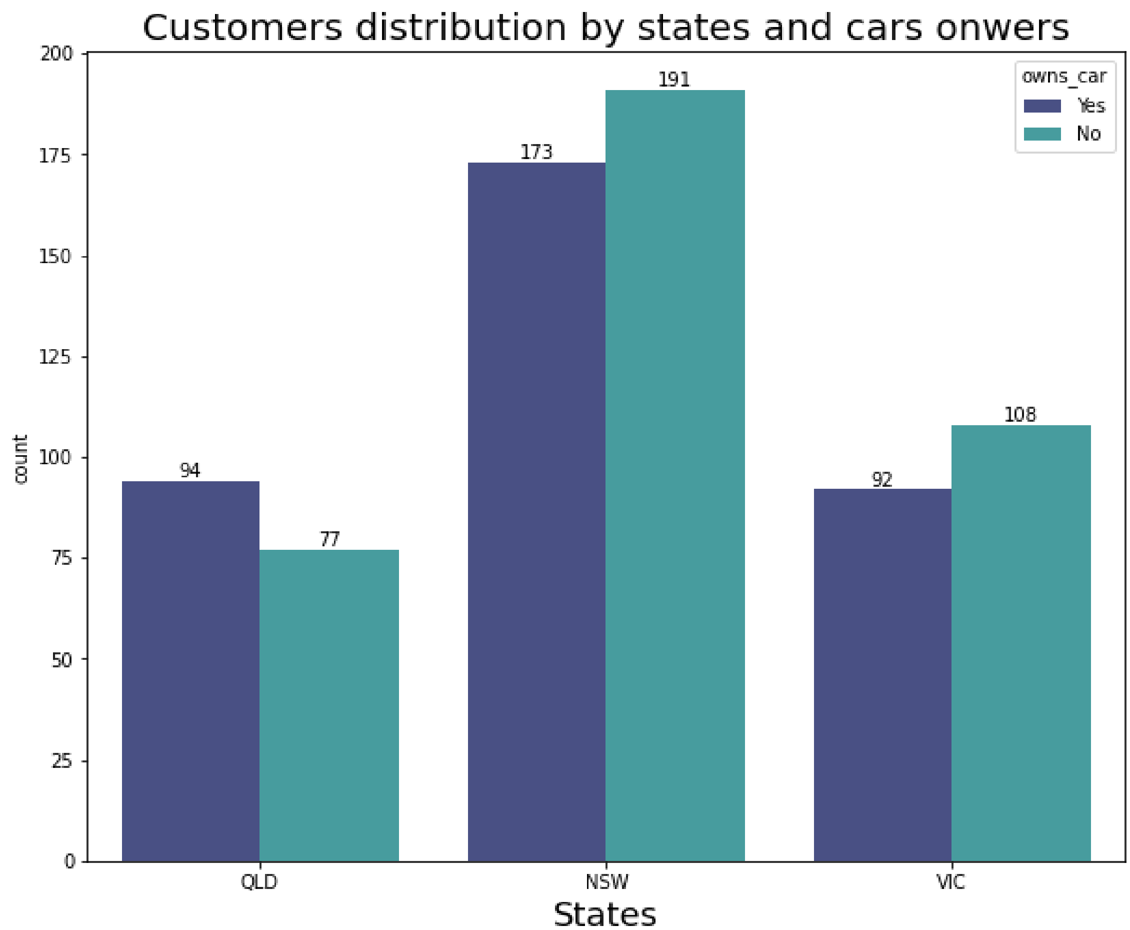


Deductions:

Clearly the NSW state has the highest customer value among the three states

- Sprocket marketting team should direct their campaigns towards their customers who live in NSW state.
- Since the other states, VIC and QLD has almost same percent of customer value, equal chances shoul be given to them

In [35]:
```
# To know which segment has the highest customer values by States and car owners

fig, ax = plt.subplots(figsize=(10, 8))


plot= sns.countplot(x='state', data=clean_set,hue = 'owns_car', ax =ax, palette= 'ma
ax.set_title('Customers distribution by states and cars onwers', fontsize=20)
ax.set_xlabel('States', fontsize =18)
for container in ax.containers:
    ax.bar_label(container)
```



Deductions:

The highest customer value in NSW state was linked to th fact that Most car owners are from there, even thoun there is a high number of people who dont have car also.

- Car owners have little significant role to adding value to customer at Sproket plt.
- Marketting campaigns sholud be lauched inrespective of whethe they own cars or not.

In [ ]:
```

```

In [ ]:
```

```

The top customers the made bike related purchase in the pas 3 years

In [64]:
```
top_sales = clean_set[clean_set['bike_purchase']>= 90]
```

In [69]:
```
top_sales.sort_values(by=['bike_purchase'], ascending=False)
```

Out[69]:

	gender	bike_purchase	DOB	job_title	job_industry_category	wealth_segment	owns_car	tenure	sta
359	Male	99	1990-07-28	Media Manager IV	Retail	High Net Worth	No	10	V
866	Female	99	1964-12-07	Dental Hygienist	Health	High Net Worth	No	14	Q
705	Female	99	1951-07-22	Cost Accountant	Financial Services	High Net Worth	No	16	NS
546	Female	99	1972-04-27	Accountant III	Financial Services	Mass Customer	No	5	NS
473	Male	99	1956-04-21	Geologist III	IT	Affluent Customer	Yes	20	NS
...	...	...	...	...	...	...	...	...	
944	Male	91	1992-08-09	Analog Circuit Design manager	Property	Mass Customer	No	5	NS
576	Male	90	1943-10-27	Sales Associate	Financial Services	Affluent Customer	No	7	V
250	Female	90	1975-03-12	Cost Accountant	Financial Services	Mass Customer	No	11	NS
136	Female	90	1990-05-29	Graphic Designer	Manufacturing	Affluent Customer	No	4	Q
943	Male	90	1974-05-28	Software Test Engineer I	Retail	Mass Customer	No	8	V

61 rows × 10 columns

Designed BY:

Faith Ubara-collins

Email: ***[ubarajf@gamil.com (mailto:ubarajf@gamil.com)](mailto:ubarajf@gamil.com)***

In [ ]:
```

```