In [ ]:
```
#import packages
```

In [1]:
```
import pandas as pd
import matplotlib.pyplot as plt
```

In [ ]:
```
#read data from csv file
```

In [10]:
```
car_prices_df = pd.read_csv("C:/Users/nikhi/Downloads/car_prices.csv/car_prices.csv")
car_prices_df .head()
```

Out[10]:

	year	make	model	trim	body	transmission	vin	state	condition	odometer	color	interior	seller
0	2015	Kia	Sorento	LX	SUV	automatic	5xyktca69fg566472	ca	5.0	16639.0	white	black	kia motors america inc
1	2015	Kia	Sorento	LX	SUV	automatic	5xyktca69fg561319	ca	5.0	9393.0	white	beige	kia motors america inc
2	2014	BMW	3 Series	328i SULEV	Sedan	automatic	wba3c1c51ek116351	ca	45.0	1331.0	gray	black	financial services remarketing (lease)
3	2015	Volvo	S60	T5	Sedan	automatic	yv1612tb4f1310987	ca	41.0	14282.0	white	black	volvo na rep/world omni
4	2014	BMW	6 Series Gran Coupe	650i	Sedan	automatic	wba6b2c57ed129731	ca	43.0	2641.0	gray	black	financial services remarketing (lease)

In [13]:
```
car_prices_df .info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 558837 entries, 0 to 558836
Data columns (total 16 columns):
 #   Column        Non-Null Count   Dtype  
---  ------        --------------   -----  
 0   year          558837 non-null  int64  
 1   make          548536 non-null  object 
 2   model         548438 non-null  object 
 3   trim          548186 non-null  object 
 4   body          545642 non-null  object 
 5   transmission  493485 non-null  object 
 6   vin           558833 non-null  object 
 7   state         558837 non-null  object 
 8   condition     547017 non-null  float64
 9   odometer      558743 non-null  float64
 10  color         558088 non-null  object 
 11  interior      558088 non-null  object 
 12  seller        558837 non-null  object 
 13  mmr           558799 non-null  float64
 14  sellingprice  558825 non-null  float64
 15  saledate      558825 non-null  object 
dtypes: float64(4), int64(1), object(11)
memory usage: 68.2+ MB
```

In [15]:
```
df = car_prices_df.copy()
df.isnull().sum()
```

Out[15]:
```
year               0
make           10301
model          10399
trim           10651
body           13195
transmission   65352
vin                4
state              0
condition      11820
odometer          94
color            749
interior         749
seller             0
mmr               38
sellingprice      12
saledate          12
dtype: int64
```

In [ ]:
```
#Exploratory Data Analysis
```

In [ ]:
```
'''Handling Missing Values In Categorical Columns
1.Fill with a Placeholder Category
2.Use Mode, Median, Mean (most frequent category)
3.Remove Null values'''
```

In [18]:
```
# Fill missing values with 'Other' category
df['make'] = df['make'].fillna('Other')
df['model'] = df['model'].fillna('Other')
df['trim'] = df['trim'].fillna('Other')
df['color'] = df['color'].fillna('Other')

# Fill missing values with mode
df['body'] = df['body'].fillna(df['body'].mode()[0])
df['transmission'] = df['transmission'].fillna(df['transmission'].mode()[0])
df['interior'] = df['interior'].fillna(df['interior'].mode()[0])

# Remove null values
df.dropna(subset=['vin'], inplace=True)
df.dropna(subset=['saledate'], inplace=True)
```

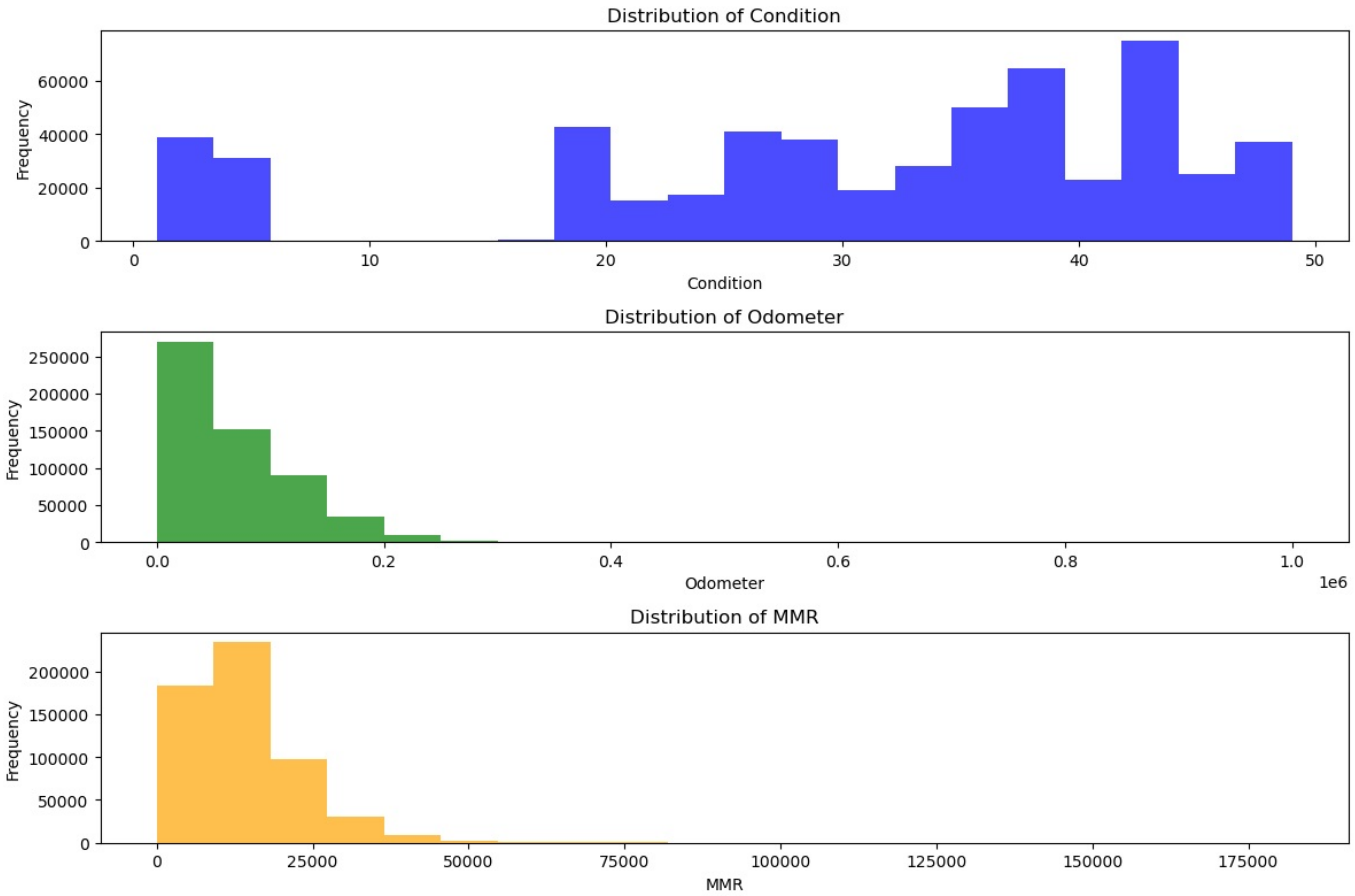
In [19]:
```
df.isnull().sum()
```

Out[19]:
```
year               0
make               0
model              0
trim               0
body               0
transmission       0
vin                0
state              0
condition      11816
odometer          94
color              0
interior           0
seller             0
mmr               22
sellingprice       0
saledate           0
dtype: int64
```

In [ ]:
```
#Handling Missing Values in Numerical Columns
```

In [20]:
```
# Assuming 'df' is your DataFrame
plt.figure(figsize=(12, 8))

# Plot histogram for 'condition'
plt.subplot(3, 1, 1)
plt.hist(df['condition'].dropna(), bins=20, color='blue', alpha=0.7)
plt.title('Distribution of Condition')
plt.xlabel('Condition')
plt.ylabel('Frequency')

# Plot histogram for 'odometer'
plt.subplot(3, 1, 2)
plt.hist(df['odometer'].dropna(), bins=20, color='green', alpha=0.7)
plt.title('Distribution of Odometer')
plt.xlabel('Odometer')
plt.ylabel('Frequency')

# Plot histogram for 'mmr'
plt.subplot(3, 1, 3)
plt.hist(df['mmr'].dropna(), bins=20, color='orange', alpha=0.7)
```

```
plt.title('Distribution of MMR')
plt.xlabel('MMR')
plt.ylabel('Frequency')

plt.tight_layout()
plt.show()
```



In [ ]:
```
#Analyzing Dataset
```

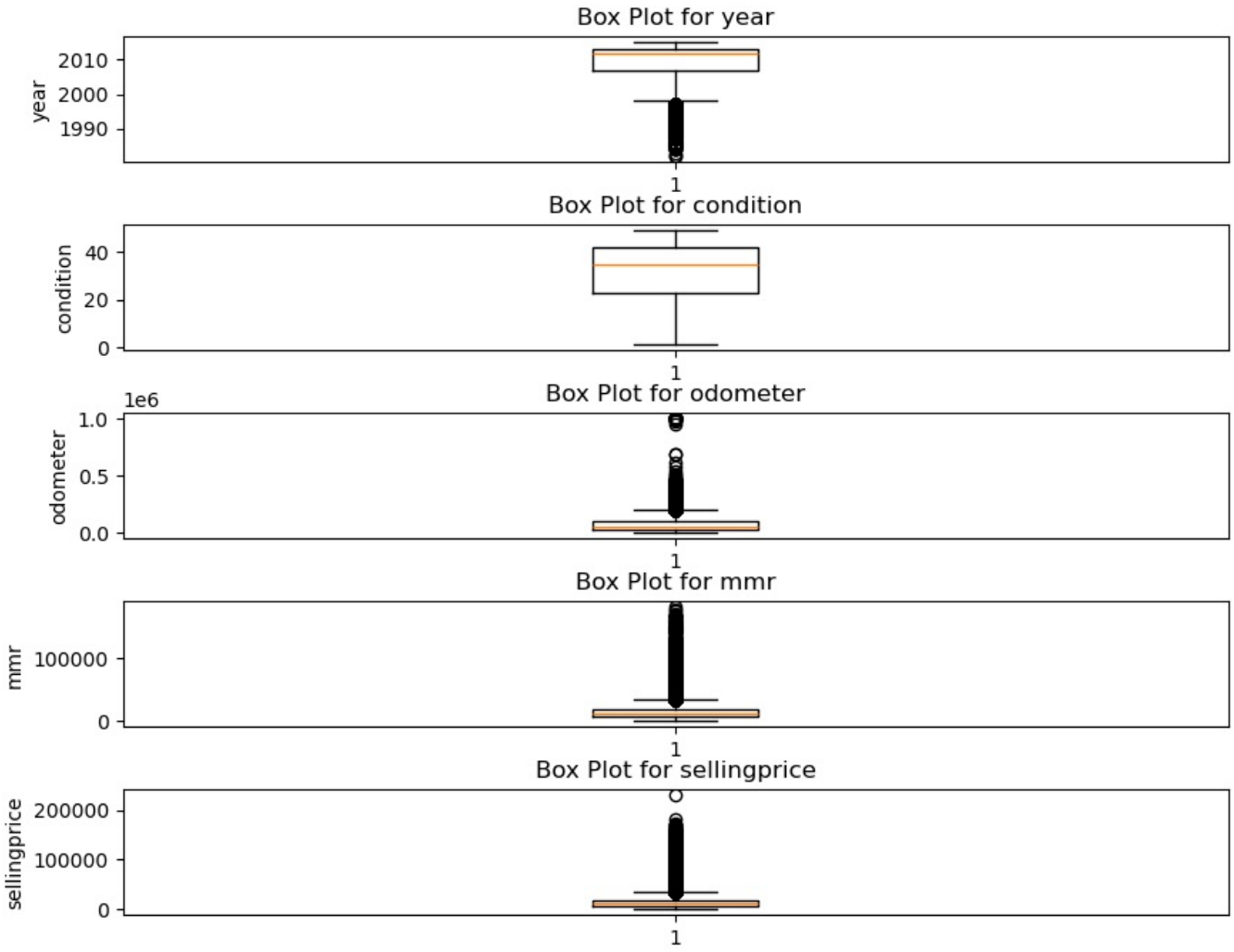
In [21]:
```
df.describe()
```

Out[21]:

	year	condition	odometer	mmr	sellingprice
count	558821.000000	547005.000000	558727.000000	558799.000000	558821.000000
mean	2010.038828	30.672557	68321.141250	13769.377495	13611.358176
std	3.966874	13.402872	53398.802013	9679.967174	9749.536466
min	1982.000000	1.000000	1.000000	25.000000	1.000000
25%	2007.000000	23.000000	28371.000000	7100.000000	6900.000000
50%	2012.000000	35.000000	52255.000000	12250.000000	12100.000000
75%	2013.000000	42.000000	99111.000000	18300.000000	18200.000000
max	2015.000000	49.000000	999999.000000	182000.000000	230000.000000

In [ ]:
```
#Outliers Handling
#Identify Outliers
```

In [22]:
```
numerical_columns = df.select_dtypes(include=['float64', 'int64']).columns

fig, axes = plt.subplots(nrows=len(numerical_columns), ncols=1, figsize=(10, 8))
fig.subplots_adjust(hspace=0.5)

for i, column in enumerate(numerical_columns):
    axes[i].boxplot(df[column].dropna())
    axes[i].set_title(f'Box Plot for {column}')
    axes[i].set_ylabel(column)

plt.show()
```



In [ ]: