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

Import the pickled DataFrames

In [2]:
```
df_orange=pd.read_pickle('orangeCounty_motorcycle')
df_chicago=pd.read_pickle('chicago_motorcycle')
df_losangeles=pd.read_pickle('losangeles_motorcycle')
df_miami=pd.read_pickle('miami_motorcycle')
df_sandiego=pd.read_pickle('sandiego_motorcycle')
df_seattle=pd.read_pickle('seattle_motorcycle')
df_newyork=pd.read_pickle('newyork_motorcycle')
df_phoenix=pd.read_pickle('phoenix_motorcycle')
df_atlanta = pd.read_pickle('atlanta_motorcycle')
df_minneapolis =pd.read_pickle('minneapolis_motorcycle')
df_boston =pd.read_pickle('boston_motorcycle')
df_portland =pd.read_pickle('portland_motorcycle')
df_lasvegas = pd.read_pickle('lasvegas_motorcycle')
df_tampa =pd.read_pickle('tampa_motorcycle')
df_dallas =pd.read_pickle('dallas_motorcycle')
df_washington =pd.read_pickle('washingtonDC_motorcycle')
df_austin =pd.read_pickle('austin_motorcycle')
df_houston =pd.read_pickle('houston_motorcycle')
df_orlando =pd.read_pickle('orlando_motorcycle')
df_philadelphia =pd.read_pickle('philadelphia_motorcycle')
df_kansascity =pd.read_pickle('kansascity_motorcycle')
df_detroit =pd.read_pickle('detroit_motorcycle')
df_charlotte =pd.read_pickle('charlotte_motorcycle')
df_stlouis =pd.read_pickle('stlouis_motorcycle')
df_northjersey =pd.read_pickle('northjersey_motorcycle')
df_pittsburgh =pd.read_pickle('pittsburgh_motorcycle')
df_southjersey =pd.read_pickle('southjersey_motorcycle')
df_columbus =pd.read_pickle('columbus_motorcycle')
df_nashville =pd.read_pickle('nashville_motorcycle')
df_baltimore =pd.read_pickle('baltimore_motorcycle')
df_boise =pd.read_pickle('boise_motorcycle')
df_spokane =pd.read_pickle('spokane_motorcycle')
df_sanantonio =pd.read_pickle('sanantonio_motorcycle')
df_sarasota =pd.read_pickle('sarasota_motorcycle')
df_milwaukee =pd.read_pickle('milwaukee_motorcycle')
df_norfolk =pd.read_pickle('norfolk_motorcycle')
df_fortmeyers =pd.read_pickle('fortmyers_motorcycle')
df_providence =pd.read_pickle('providence_motorcycle')
df_indianapolis =pd.read_pickle('indianapolis_motorcycle')
df_jacksonville =pd.read_pickle('jacksonville_motorcycle')
df_cincinnati =pd.read_pickle('cincinnati_motorcycle')
```

Concatane them into one dataframe and reset the index

In [3]:
```
df =pd.concat([df_orange, df_chicago, df_losangeles, df_miami, df_sandiego,
               df_phoenix, df_atlanta, df_minneapolis, df_boston, df_portla
               df_washington, df_austin, df_houston, df_orlando, df_philade
               df_charlotte, df_stlouis, df_northjersey, df_pittsburgh, df_
               df_baltimore, df_boise, df_spokane, df_sanantonio, df_saraso
               df_providence, df_indianapolis, df_jacksonville, df_cincinna
              ])
df =df.reset_index(drop=True)
df
```

Out[3]:

	Price	URL	title	bikeTitle	
0	$13,995	https://orangecounty.craigslist.org/mcd/d/oran...	2002 Harley-Davidson FLSTSI SKU:12858	2002 Harley-Davidson FLSTSI	1HD1BYB19 fuel: , odor
1	$9,995	https://orangecounty.craigslist.org/mcd/d/oran...	2003 Harley-Davidson FLHTCUI (ANNIVERSARY) SKU...	2003 Harley-Davidson FLHTCUI (ANNIVER	1HD1FCW1X fuel: , odor
2	$22,995	https://orangecounty.craigslist.org/mcd/d/oran...	2019 Harley-Davidson FLHX - Street Glide SKU:1...	2019 Harley-Davidson FLHX - Street Gl	1HD1KBC3X fuel: , odor
3	$3,000	https://orangecounty.craigslist.org/mcy/d/oran...	2002 Honda VTX 1800 C Custom	2002 Honda VTX	, condition engine dis
4	$19,995	https://orangecounty.craigslist.org/mcd/d/oran...	2016 Harley-Davidson FLHX - Street Glide SKU:1...	2016 Harley-Davidson FLHX - Street Gl	1HD1KBM30 fuel: , odor
...	...	...	...	...	
23039	$15,000	https://cincinnati.craigslist.org/mcy/d/cincin...	Harley Panhead- trade	Harley Davidson Panhead	, fuel: gas, 1000, tra
23040	$7,000	https://cincinnati.craigslist.org/mcy/d/cincin...	2012 Kawasaki Vulcan 900 Classic	2012 kawasaki vulcan 900 classic	, condition engine dis
23041	$8,000	https://cincinnati.craigslist.org/mcy/d/hamilt...	2013 Triumph Tiger Explorer 1200 ABS	triumph tiger explorer abs	, condition engine dis
23042	$1,500	https://cincinnati.craigslist.org/mcy/d/floren...	2006 Genuine Black Cat 🐈‍⬛ 50cc 2 stroke motor...	2006 Genuine Black Cat 50	, conditio engine dis
23043	$3,500	https://cincinnati.craigslist.org/mcd/d/cincin...	1994 SUZUKI DR350SE W/8K MILES	1994 suzuki dr350se	, condition delivery ava

23044 rows × 16 columns

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

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 23044 entries, 0 to 23043
Data columns (total 16 columns):
 #   Column                    Non-Null Count  Dtype 
---  ------                    --------------  ----- 
 0   Price                     23044 non-null  object
 1   URL                       23044 non-null  object
 2   title                     23044 non-null  object
 3   bikeTitle                 23044 non-null  object
 4   attributes                23044 non-null  object
 5   VIN                       7695 non-null   object
 6   fuel                      21332 non-null  object
 7   odometer                  20995 non-null  object
 8   paint color               14857 non-null  object
 9   title status              18122 non-null  object
 10  transmission              21374 non-null  object
 11  condition                 12106 non-null  object
 12  engine displacement (CC)  8369 non-null   object
 13  type                      9423 non-null   object
 14  year                      16766 non-null  object
 15  years                     20156 non-null  object
dtypes: object(16)
memory usage: 2.8+ MB
```

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

Out[5]:

	Price	URL	title	bikeTitle	attributes	VIN	
count	23044	23044	23044	23044	23044	7695	2
unique	2000	22651	17407	13444	17507	5456	
top	$0	https://seattle.craigslist.org/oly/mcd/d/olymp...	Harley Davidson	harley davidson	, fuel: gas, odometer: 1, transmission: manual	NA	
freq	644	12	56	121	163	36	1

In [6]:
```
#Make the title column lowercase before we get each company
df['title_lower'] = df['title'].str.lower()
```

Get the company for each motorcycle listing

Made a list of the top company motorcycles, and then if the company matches with some string in the title it will be extracted

In [7]:
```
s =['harley-davidson', 'honda', 'yamaha', 'ducati', 'suzuki','ktm', 'bmw',

matcher(x):
for i in makes:
    if i.lower() in x.lower():
        return i
else:
    return np.nan
```

In [8]:
```
#applying the function and creating a new column
df['make'] = df['title'].apply(matcher)
```

In [9]:
```
#checking if we get something different or additional information that wasn
df['make_title'] = df['bikeTitle'].apply(matcher)
```

In [10]:
```
df['make'].value_counts().sum()
```

Out[10]: 15590

In [11]:
```
df['make_title'].value_counts().sum()
```

Out[11]: 16150

As you can see we have different amounts.

We will then fill in all the missing values in make and make_title with the other one and create a new column

In [12]:
```
df['final_make'] = df["make"].fillna(df["make_title"])
```

In [13]:
```
df['final_make'] = df["make_title"].fillna(df["make"])
```

In [14]:
```
#it worked
df['final_make'].value_counts()
```

Out[14]:
```
harley-davidson    3507
honda              3322
yamaha             2338
kawasaki           2205
suzuki             1390
bmw                1029
ducati              785
ktm                 764
triumph             681
indian              425
victory             194
aprilia             138
vespa               125
norton                5
bajaj                 1
Name: final_make, dtype: int64
```

Doing the same thing for year and years and making a new column

In [15]:
```
df['year'].value_counts().sum()
```

Out[15]:
```
16766
```

In [16]:
```
df['years'].value_counts().sum()
```

Out[16]:
```
20156
```

In [17]:
```
df['final_years'] = df["year"].fillna(df["years"])
```

In [18]:
```
df['final_years'] = df["years"].fillna(df["year"])
```

In [19]:
```
df['final_years'].value_counts().sum()
```

Out[19]:
```
21995
```

In [20]:
```
df['paint color'].value_counts()
```

Out[20]:
```
black                 3316
red                   1756
blackstreet legal     1300
custom                1206
blue                  1191
white                  825
redstreet legal        772
bluestreet legal       558
silver                 515
grey                   477
orange                 459
green                  385
whitestreet legal      314
                       268
customstreet legal     236
silverstreet legal     212
orangestreet legal     203
greystreet legal       196
yellow                 181
greenstreet legal      155
yellowstreet legal      96
brown                   93
purple                  52
purplestreet legal      51
brownstreet legal       40
Name: paint color, dtype: int64
```

Cleaning up the paint color column to have a uniform color

In [21]:
```
df['paint color'] = df['paint color'].str.replace('blackstreet legal', 'bla
df['paint color'] = df['paint color'].str.replace('redstreet legal', 'red')
df['paint color'] = df['paint color'].str.replace('bluestreet legal', 'blue
df['paint color'] = df['paint color'].str.replace('whitestreet legal', 'whi
df['paint color'] = df['paint color'].str.replace('customstreet legal', 'cu
df['paint color'] = df['paint color'].str.replace('orangestreet legal', 'or
df['paint color'] = df['paint color'].str.replace('greenstreet legal', 'gre
df['paint color'] = df['paint color'].str.replace('greystreet legal', 'grey
df['paint color'] = df['paint color'].str.replace('yellowstreet legal', 'ye
df['paint color'] = df['paint color'].str.replace('purplestreet legal', 'pu
df['paint color'] = df['paint color'].str.replace('brownstreet legal', 'bro
df['paint color'] = df['paint color'].str.replace('silverstreet legal', 'si
```

In [22]:
```
df
```

Out[22]:

	Price	URL	title	bikeTitle	
0	$13,995	https://orangecounty.craigslist.org/mcd/d/oran...	2002 Harley-Davidson FLSTSI SKU:12858	2002 Harley-Davidson FLSTSI	1HD1BYB19 fuel: , odor
1	$9,995	https://orangecounty.craigslist.org/mcd/d/oran...	2003 Harley-Davidson FLHTCUI (ANNIVERSARY) SKU...	2003 Harley-Davidson FLHTCUI (ANNIVER	1HD1FCW1X fuel: , odor
2	$22,995	https://orangecounty.craigslist.org/mcd/d/oran...	2019 Harley-Davidson FLHX - Street Glide SKU:1...	2019 Harley-Davidson FLHX - Street Gl	1HD1KBC3X fuel: , odor
3	$3,000	https://orangecounty.craigslist.org/mcy/d/oran...	2002 Honda VTX 1800 C Custom	2002 Honda VTX	, condition engine dis
4	$19,995	https://orangecounty.craigslist.org/mcd/d/oran...	2016 Harley-Davidson FLHX - Street Glide SKU:1...	2016 Harley-Davidson FLHX - Street Gl	1HD1KBM30 fuel: , odor
...	...	...	...	...	
23039	$15,000	https://cincinnati.craigslist.org/mcy/d/cincin...	Harley Panhead- trade	Harley Davidson Panhead	, fuel: gas, 1000, tra
23040	$7,000	https://cincinnati.craigslist.org/mcy/d/cincin...	2012 Kawasaki Vulcan 900 Classic	2012 kawasaki vulcan 900 classic	, condition engine dis
23041	$8,000	https://cincinnati.craigslist.org/mcy/d/hamilt...	2013 Triumph Tiger Explorer 1200 ABS	triumph tiger explorer abs	, condition engine dis
23042	$1,500	https://cincinnati.craigslist.org/mcy/d/floren...	2006 Genuine Black Cat 🐈‍⬛ 50cc 2 stroke motor...	2006 Genuine Black Cat 50	, conditio engine dis
23043	$3,500	https://cincinnati.craigslist.org/mcd/d/cincin...	1994 SUZUKI DR350SE W/8K MILES	1994 suzuki dr350se	, condition delivery ava

23044 rows × 21 columns

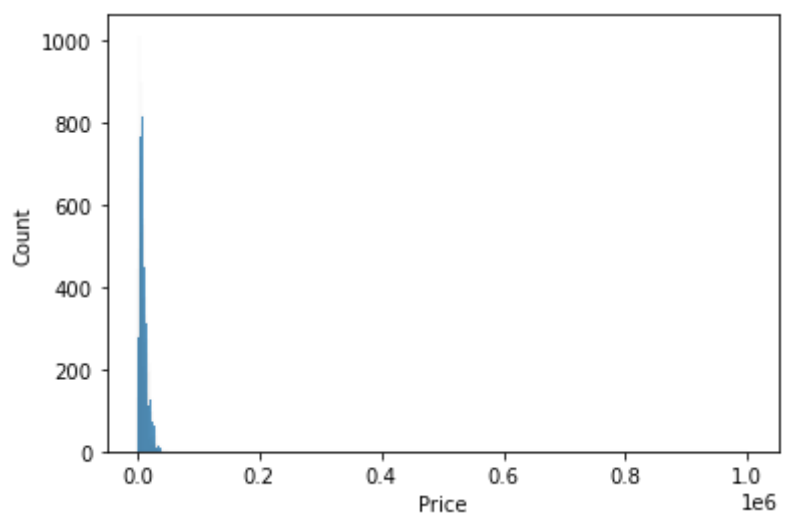
Pickling the dataFrame to be used for the models

In [23]:
```
# df.to_pickle('df_model')
```

Last minute cleaning

In [24]:
```
#dropping null values in necessary columns
df = df.dropna(subset =['final_years', 'final_make'])
#dropping duplicate columns and unnecessary columns
df =df.drop(['make','make_title', 'year', 'years', 'title_lower', 'URL', 'b
#changing the years column to a number value instead of string
df = df[df['final_years'].astype(str).str.isdigit()]
#getting just the numbers out of the odometer column(no commas etc..)
df['odometer'] = df.odometer.str.extract('(^\d*)')
df[["final_years"]] = df[['final_years']].apply(pd.to_numeric)
#cleaning price to make it an integer and making it an integer
df['Price']= df['Price'].apply(lambda x: x.replace('$',''))
df['Price']= df['Price'].apply(lambda x: x.replace(',',''))
df[['Price']] = df[['Price']].apply(pd.to_numeric)
df[['odometer']] = df[['odometer']].apply(pd.to_numeric)
df =df.reset_index(drop=True)
```

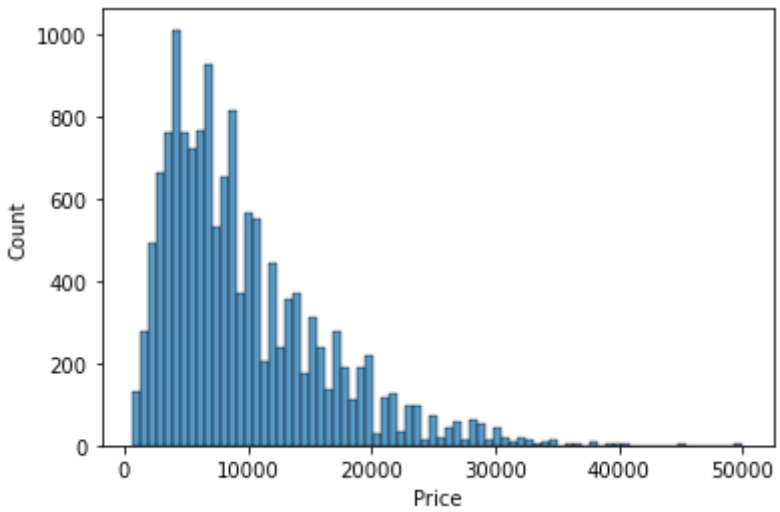
In [25]:
```
sns.histplot(df['Price'])
plt.ticklabel_format(style='plain', axis='y')
```



We can see that there are outliers in our price so we need to trim the data. There shouldn't be any motorcycles on craigslist being sold over $50,000

In [26]:
```
df = df[df['Price']<50000]
df=df[df['Price']>600]
```

In [27]:
```
sns.histplot(df['Price'])
plt.ticklabel_format(style='plain', axis='y')
```

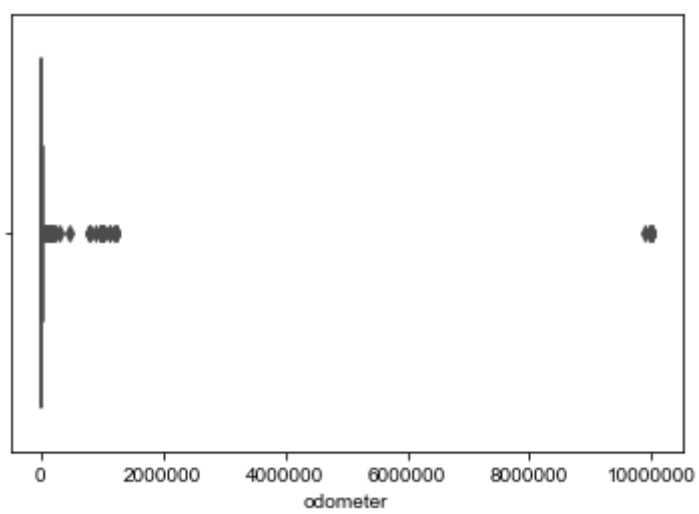


We can see the data is skewed to the right but this looks workable

check Odometer

In [28]:
```
df['odometer']= df['odometer'].astype(float)
```

In [29]:
```
plt.ticklabel_format(style='plain')
sns.set_theme(style="whitegrid")
ax = sns.boxplot(x=df["odometer"])
```

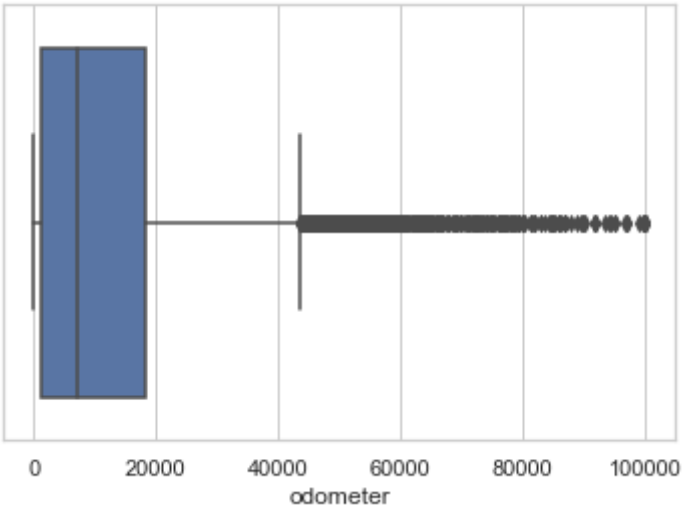


We can see some outliers for the milieage

Going to use motorcycles under 100,000 miles

In [30]:
```
df = df[df['odometer']<100000]
```

In [31]:
```
plt.ticklabel_format(style='plain')
sns.set_theme(style="whitegrid")
ax = sns.boxplot(x=df["odometer"])
```



This looks a lot better

Let's check how the years are

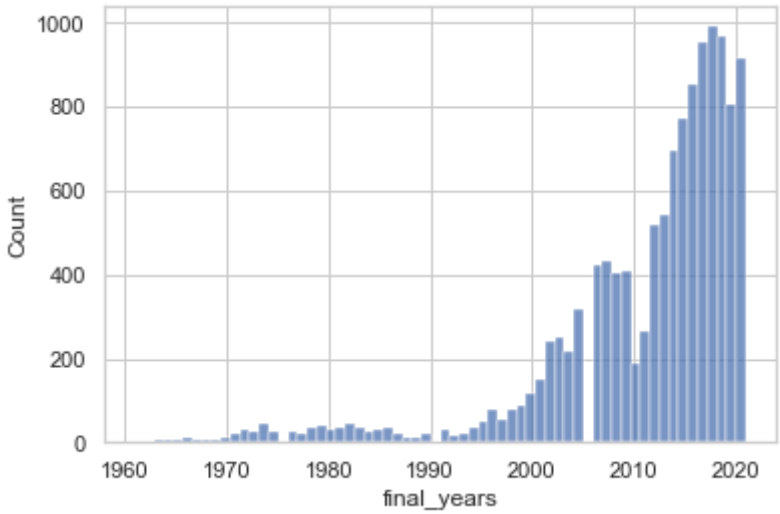
In [32]:
```
df['final_years'].sort_values()
```

Out[32]:
```
2953          2
2613          2
714           3
2930          3
1551          3
         ...
9130       2022
2789       2022
6347       2022
9324       2022
1006    1502021
Name: final_years, Length: 13055, dtype: int64
```

definitely some outliers

In [33]:
```
df = df[df['final_years']>1960]
df = df[df['final_years']<2022]
```

In [34]:
```
sns.histplot(df['final_years'])
plt.ticklabel_format(style='plain', axis='y')
```

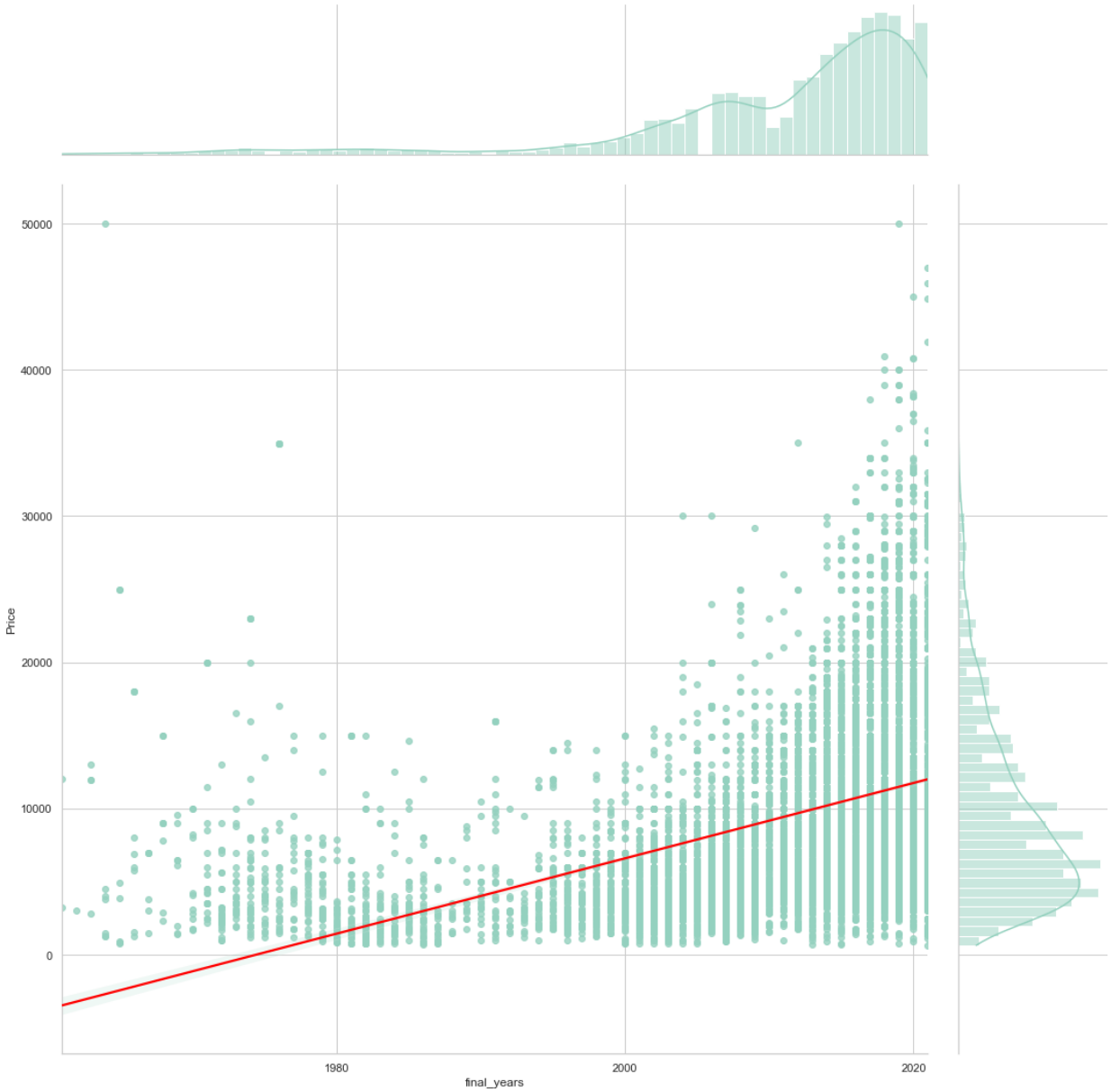


Showing the distribution of year of motorcycle tied to the price

In [35]:

```
sns.set_palette("GnBu_d")
sns.set_style('whitegrid')
g =sns.jointplot(x='final_years',y='Price',data=df,size =15, kind='reg')
sns.set_context("talk", font_scale=3)
regline = g.ax_joint.get_lines()[0]
regline.set_color('red')
regline.set_zorder(5)
```

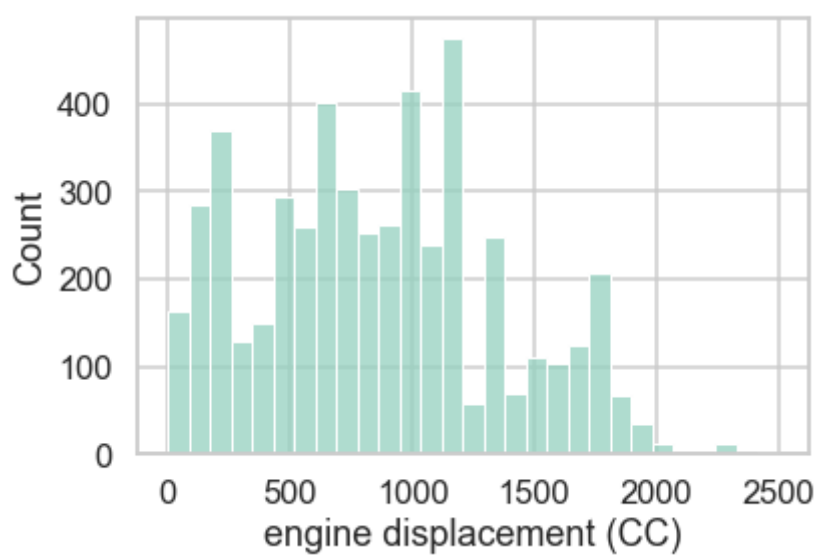
```
/Users/avijames/anaconda3/lib/python3.8/site-packages/seaborn/axisgrid.p
y:2073: UserWarning: The `size` parameter has been renamed to `height`; p
lease update your code.
  warnings.warn(msg, UserWarning)
```



The graph shows that higher price is correlated with higher year but just because it is a higher year does not mean it is a higher price

Distribution of engine displacement

In [36]:
```
sns.set_context("talk", font_scale=1)
df['engine displacement (CC)']= df['engine displacement (CC)'].astype(float
sns.histplot(df['engine displacement (CC)'])
plt.ticklabel_format(style='plain', axis='y')
```



In [37]:
```
df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
Int64Index: 12484 entries, 0 to 14925
Data columns (total 13 columns):
 #   Column                    Non-Null Count  Dtype  
---  ------                    --------------  -----  
 0   Price                     12484 non-null  int64  
 1   title                     12484 non-null  object 
 2   VIN                       5504 non-null   object 
 3   fuel                      12357 non-null  object 
 4   odometer                  12484 non-null  float64
 5   paint color               9428 non-null   object 
 6   title status              11689 non-null  object 
 7   transmission              12403 non-null  object 
 8   condition                 6932 non-null   object 
 9   engine displacement (CC)  5045 non-null   float64
 10  type                      5508 non-null   object 
 11  final_make                12484 non-null  object 
 12  final_years               12484 non-null  int64  
dtypes: float64(2), int64(2), object(9)
memory usage: 1.3+ MB
```

In [38]:
```
new_df =pd.DataFrame()
```

In [39]:
```
new_df['bike_price_median']= df.groupby('final_make')['Price'].median().sor
```

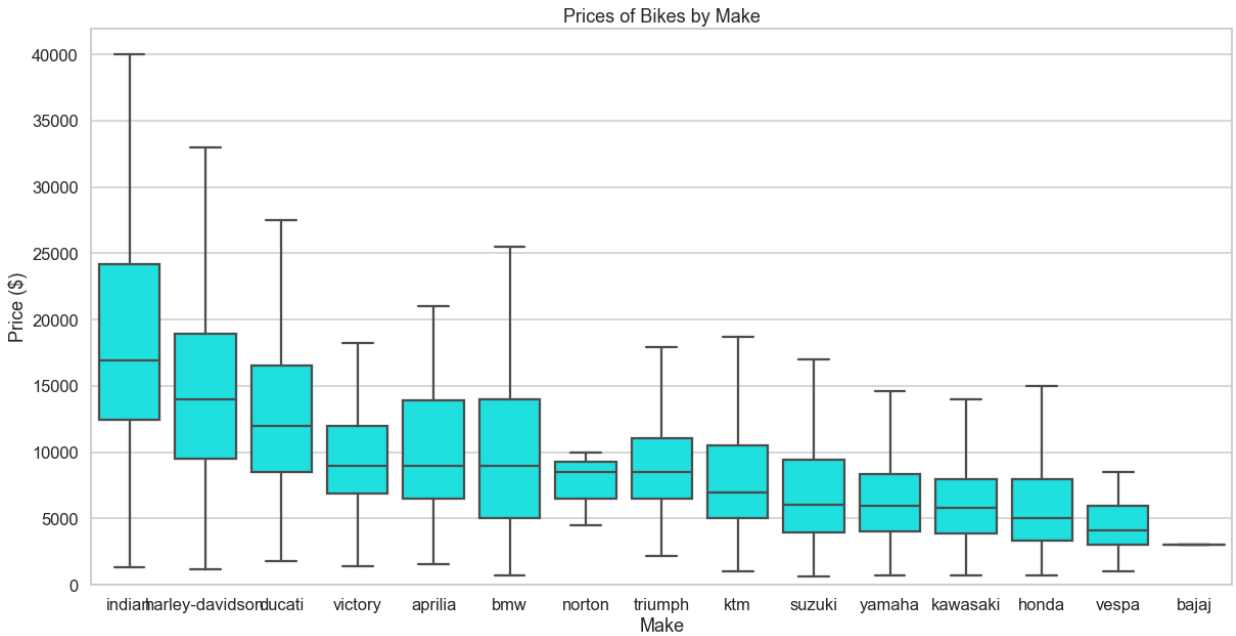
In [40]:
```
df.groupby('final_make')['Price'].median().sort_values(ascending=False)
```

Out[40]:
```
final_make
indian             16950.0
harley-davidson    13991.0
ducati             11995.0
victory             8999.0
aprilia             8995.0
bmw                 8995.0
norton              8500.0
triumph             8499.5
ktm                 6999.0
suzuki              6000.0
yamaha              5999.0
kawasaki            5799.0
honda               4999.0
vespa               4099.0
bajaj               3000.0
Name: Price, dtype: float64
```

In [41]:
```
boxplot_order =new_df['bike_price_median']
```

In [42]:
```
plt.figure(figsize=(20,10))
# Create a boxplot for each state:
sns.boxplot(x='final_make',y='Price',
            data=df,
            showfliers=False,order=boxplot_order, color = 'Aqua')

# Set labels and axis limits:
plt.ylim(0,);
plt.xlabel('Make')
plt.ylabel('Price ($)')
plt.ticklabel_format(style='plain', axis='y')
plt.title('Prices of Bikes by Make');

# Print maximum mean and median value
print('Maximum median',':',df.groupby('final_make').mean().max())
print('Maximum mean',':',df.groupby('final_make').mean().median())
```

```
Maximum median : Price                       18446.607143
odometer                    23923.482759
engine displacement (CC)     1479.145833
final_years                  2018.126623
dtype: float64
Maximum mean : Price                       8081.087719
odometer                    10519.280000
engine displacement (CC)      825.004228
final_years                  2011.461538
dtype: float64
```



In [43]:
```
# df.drop("VIN", inplace =True, axis =1)
df =df.reset_index(drop=True)
df['condition'] = df['condition'].str.replace('excellentcryptocurrency ok',
df['condition'] = df['condition'].str.replace('like newcryptocurrency ok',
df['condition'] = df['condition'].str.replace('goodcryptocurrency ok', 'goo
df['condition'] = df['condition'].str.replace('faircryptocurrency ok', 'fai
df['condition'] = df['condition'].str.replace('newcryptocurrency ok', 'new'
df['fuel'] = df['fuel'].fillna(value='other')
df['paint color'] = df['paint color'].fillna(value='other')
df['title status'] = df['title status'].fillna(value='other')
df['transmission'] = df['transmission'].fillna(value='other')
df['condition'] = df['condition'].fillna(value='other')
# df['engine displacement (CC)'] = df['engine displacement (CC)'].fillna(va
df['type'] = df['type'].fillna(value='other')
# df = df[df['engine displacement (CC)']!='other']
```

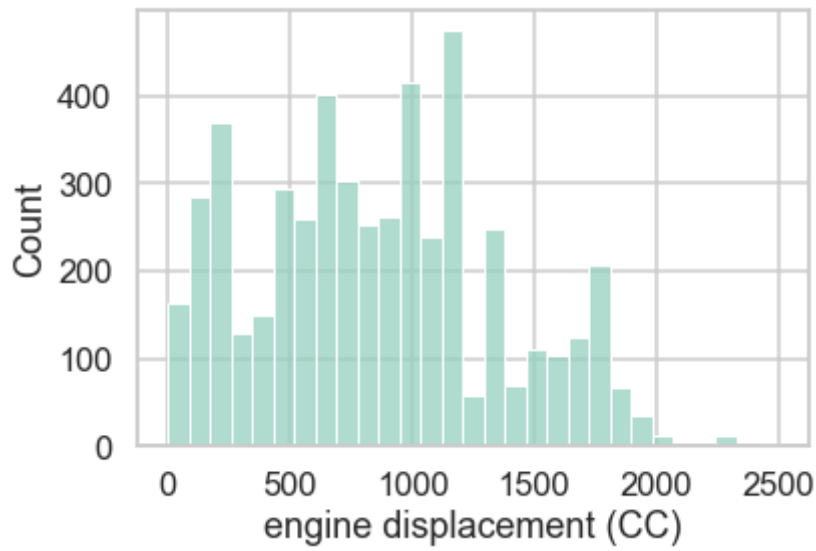
In [44]:
```
df
```

Out[44]:

	Price	title	VIN	fuel	odometer	paint color	title status	transmission
0	13995	2002 Harley-Davidson FLSTSI SKU:12858	1HD1BYB192Y026148		12382.0	black	clean	manual
1	9995	2003 Harley-Davidson FLHTCUI (ANNIVERSARY) SKU...	1HD1FCW1X3Y604180		36852.0	blue	clean	manual
2	22995	2019 Harley-Davidson FLHX - Street Glide SKU:1...	1HD1KBC3XKB603737		11704.0	custom	clean	manual
3	3000	2002 Honda VTX 1800 C Custom	NaN	gas	69000.0	black	clean	manual
4	19995	2016 Harley-Davidson FLHX - Street Glide SKU:1...	1HD1KBM30GB657117		9284.0	black	clean	manual
...	...	...	...	...	...	...	...	...
12479	9000	1999 Ultra Classic Harley Davidson	NaN	gas	13000.0	green	clean	manual
12480	19995	'20 BMW R1250GSA	NaN	gas	11715.0	green	clean	manual
12481	3200	2004 V-Strom 1000	NaN	gas	42000.0	blue	clean	manual
12482	7000	2012 Kawasaki Vulcan 900 Classic	NaN	gas	17543.0	black	clean	manual
12483	3500	1994 SUZUKI DR350SE W/8K MILES	NaN	gas	8100.0	blue	clean	manual

12484 rows × 13 columns

In [45]:
```
df['engine displacement (CC)']= df['engine displacement (CC)'].astype(float
sns.histplot(df['engine displacement (CC)'])
plt.ticklabel_format(style='plain', axis='y')
```



Putting the engine displacements into bins so that we can categorize them better

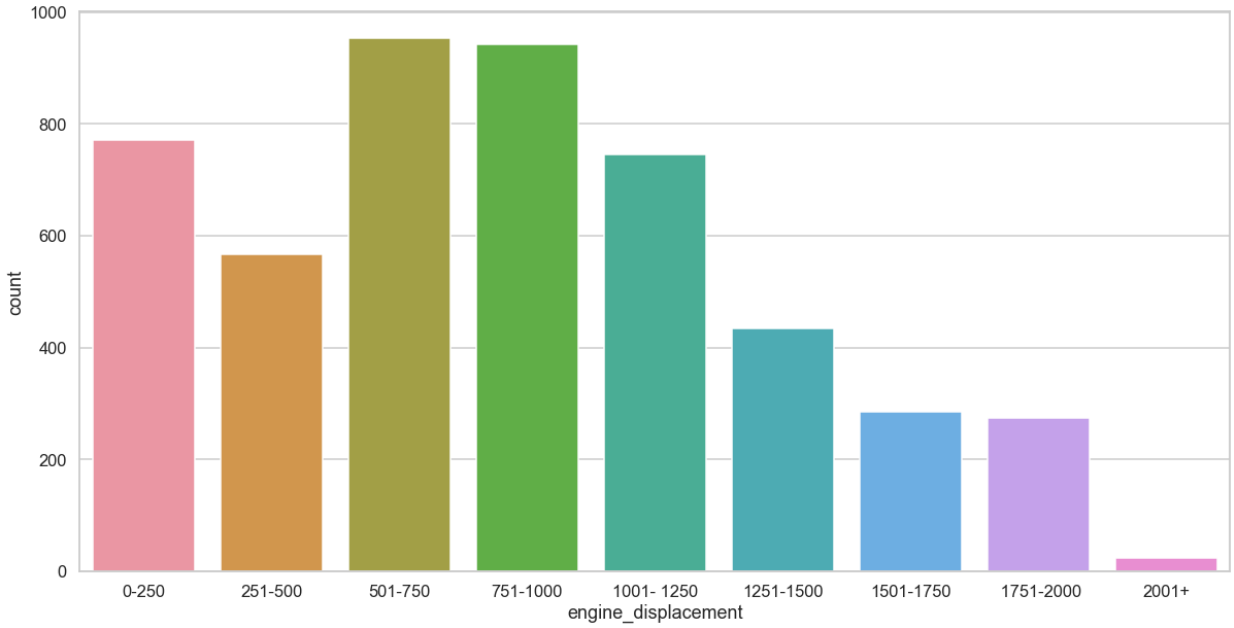
In [46]:
```
bins = [0,250,500,750,1000,1250,1500,1750,2000, np.inf]
names =['0-250', '251-500', '501-750', '751-1000', '1001- 1250', '1251-1500

df['engine_displacement'] = pd.cut(df['engine displacement (CC)'], bins, la
```

In [47]:
```
df['engine_displacement'].value_counts().sum()
```

Out[47]: 4999

In [48]:
```
plt.figure(figsize=(20,10))
sns.countplot(df['engine_displacement'])
plt.ticklabel_format(style='plain', axis='y')
```

```
/Users/avijames/anaconda3/lib/python3.8/site-packages/seaborn/_decorator
s.py:36: FutureWarning: Pass the following variable as a keyword arg: x.
From version 0.12, the only valid positional argument will be `data`, and
passing other arguments without an explicit keyword will result in an err
or or misinterpretation.
  warnings.warn(
```



In [49]:
```
df['engine_displacement']
```

Out[49]:
```
0               NaN
1               NaN
2               NaN
3         1751-2000
4               NaN
             ...
12479           NaN
12480    1001- 1250
12481      751-1000
12482      751-1000
12483       251-500
Name: engine_displacement, Length: 12484, dtype: category
Categories (9, object): ['0-250' < '251-500' < '501-750' < '751-1000' ...
'1251-1500' < '1501-1750' < ' 1751-2000' < '2001+']
```

In [50]:
```
# df.to_pickle('df_model')
```