Practice seaborn and matplotlib

In [1]:
```
# import library
import pandas as pd
import matplotlib.pyplot as plt
import seaborn as sns
%matplotlib inline
%config InlineBackend.figure_format = 'retina'
```

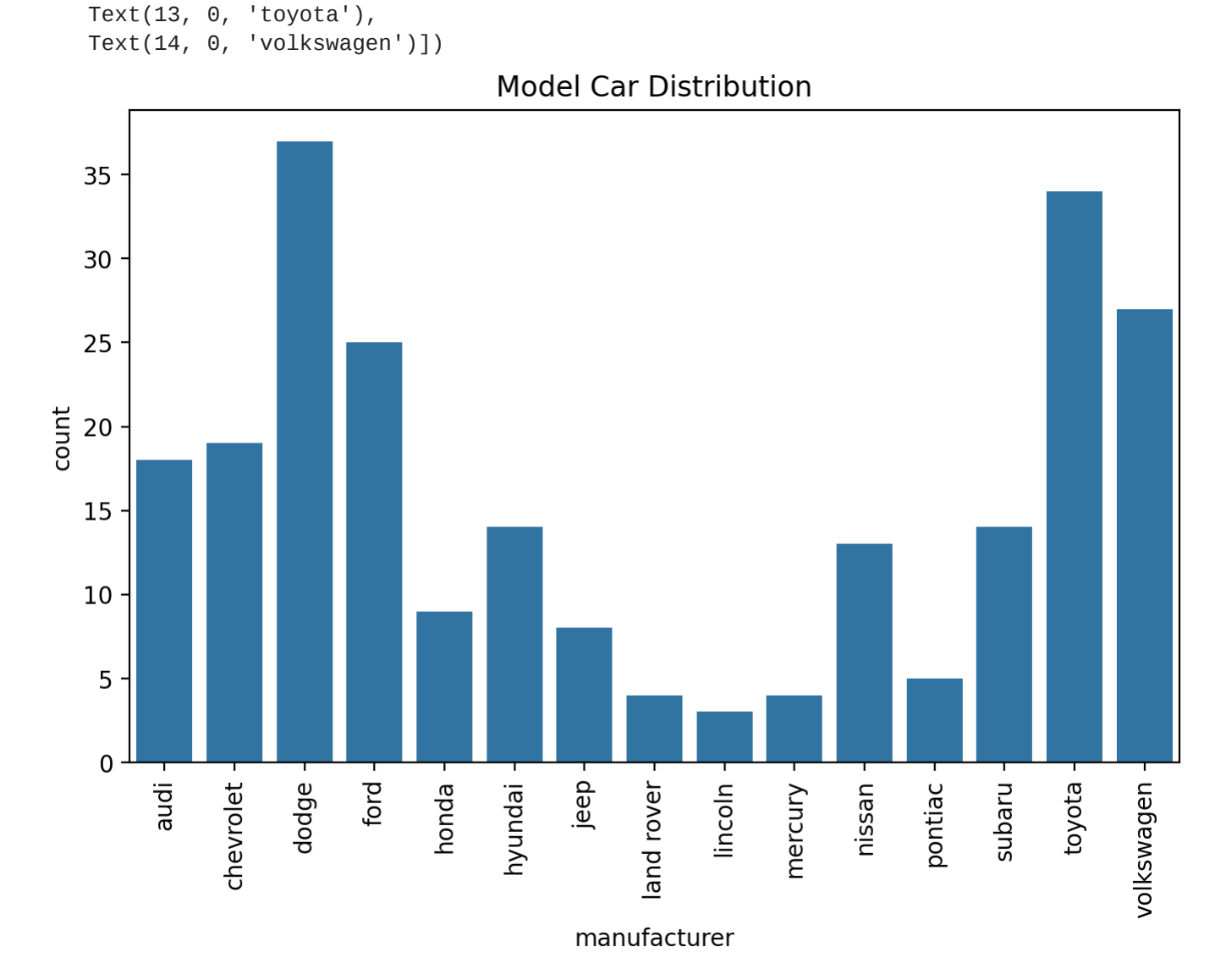
In [3]:
```
# import data
url = 'https://raw.githubusercontent.com/prasertcbs/tutorial/master/mpg.csv'
df = pd.read_csv(url)
df.head()
```

Out[3]:

	manufacturer	model	displ	year	cyl	trans	drv	cty	hwy	fl	class
0	audi	a4	1.8	1999	4	auto(l5)	f	18	29	p	compact
1	audi	a4	1.8	1999	4	manual(m5)	f	21	29	p	compact
2	audi	a4	2.0	2008	4	manual(m6)	f	20	31	p	compact
3	audi	a4	2.0	2008	4	auto(av)	f	21	30	p	compact
4	audi	a4	2.8	1999	6	auto(l5)	f	16	26	p	compact

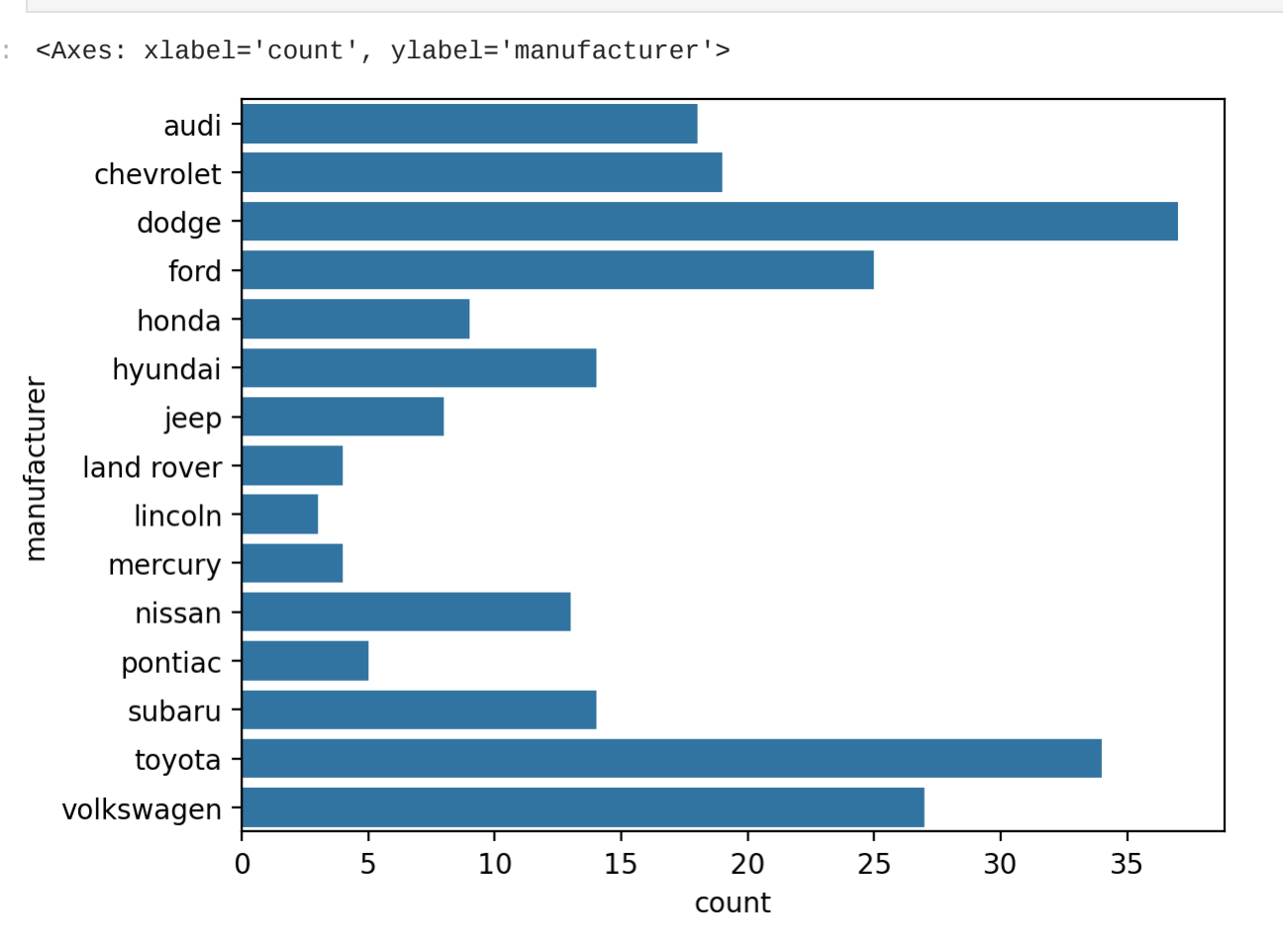
In [5]:
```
# find how many audi car model
plt.figure(figsize=(8,5)) # customize chart size >> figure(figsize=(x,y))
plt.title('Model Car Distribution')
sns.countplot(x='manufacturer', data=df)
plt.xticks(rotation=90) # rotation column name in x axis 90
```

Out[5]:
```
([0, 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14],
 [Text(0, 0, 'audi'),
  Text(1, 0, 'chevrolet'),
  Text(2, 0, 'dodge'),
  Text(3, 0, 'ford'),
  Text(4, 0, 'honda'),
  Text(5, 0, 'hyundai'),
  Text(6, 0, 'jeep'),
  Text(7, 0, 'land rover'),
  Text(8, 0, 'lincoln'),
  Text(9, 0, 'mercury'),
  Text(10, 0, 'nissan'),
  Text(11, 0, 'pontiac'),
  Text(12, 0, 'subaru'),
  Text(13, 0, 'toyota'),
  Text(14, 0, 'volkswagen')])
```



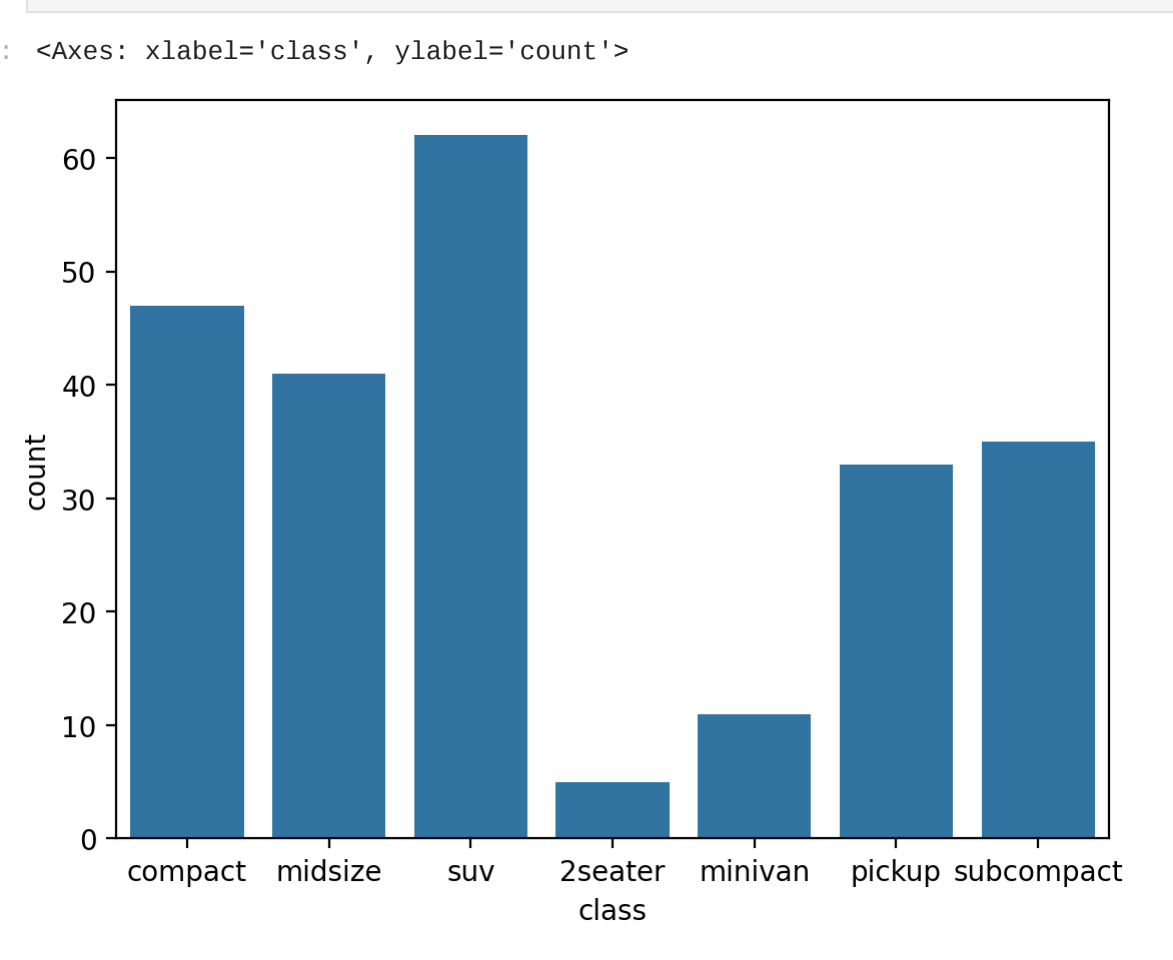
In [6]:
```
sns.countplot(y='manufacturer', data=df)
```

Out[6]: <Axes: xlabel='count', ylabel='manufacturer'>



In [7]:
```
sns.countplot(x='class', data=df)
```

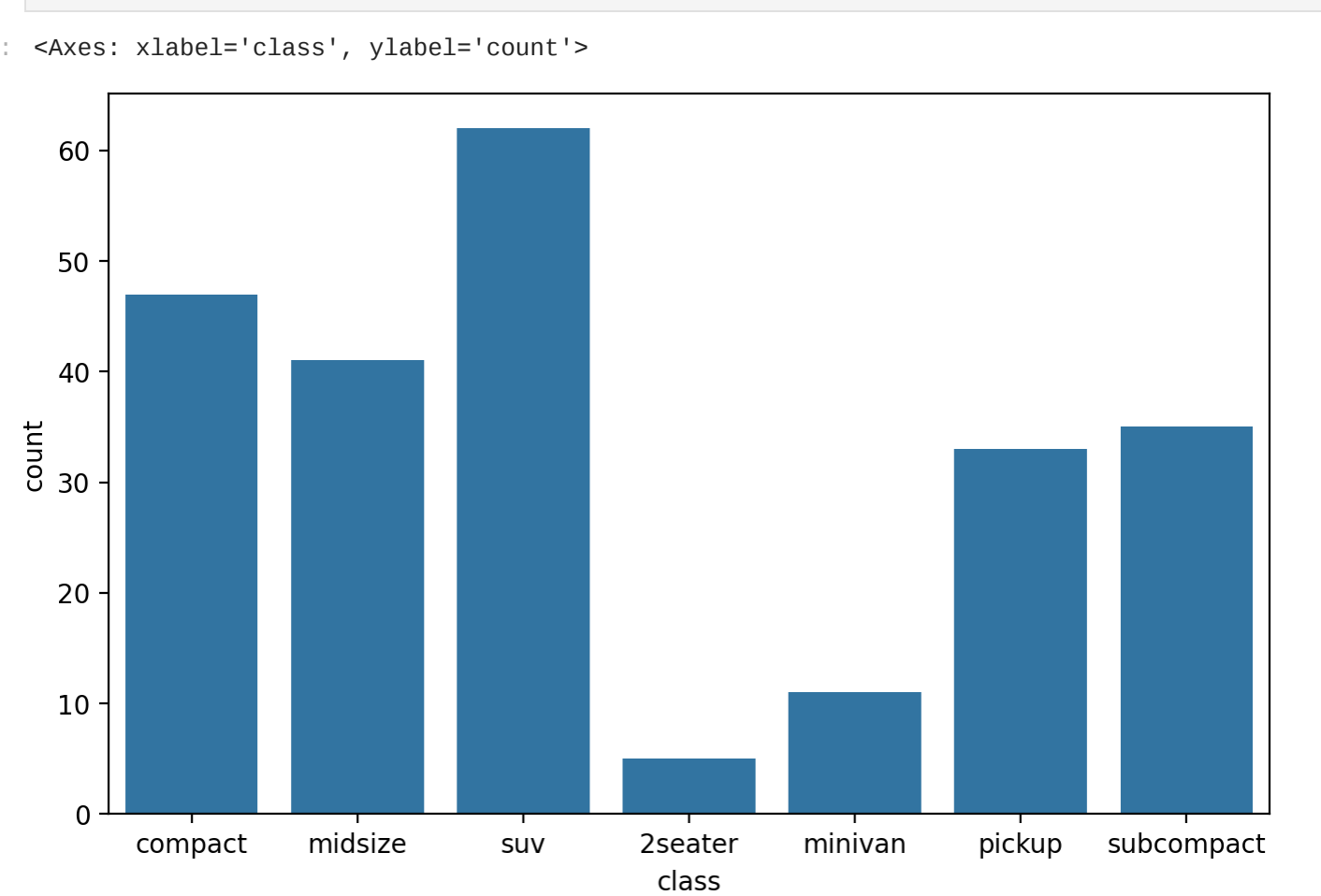
Out[7]: <Axes: xlabel='class', ylabel='count'>



set cahrt size

In [8]:
```
plt.figure(figsize=(8,5)) # customize chart size >> figure(figsize=(x,y))
sns.countplot(x='class', data=df)
```

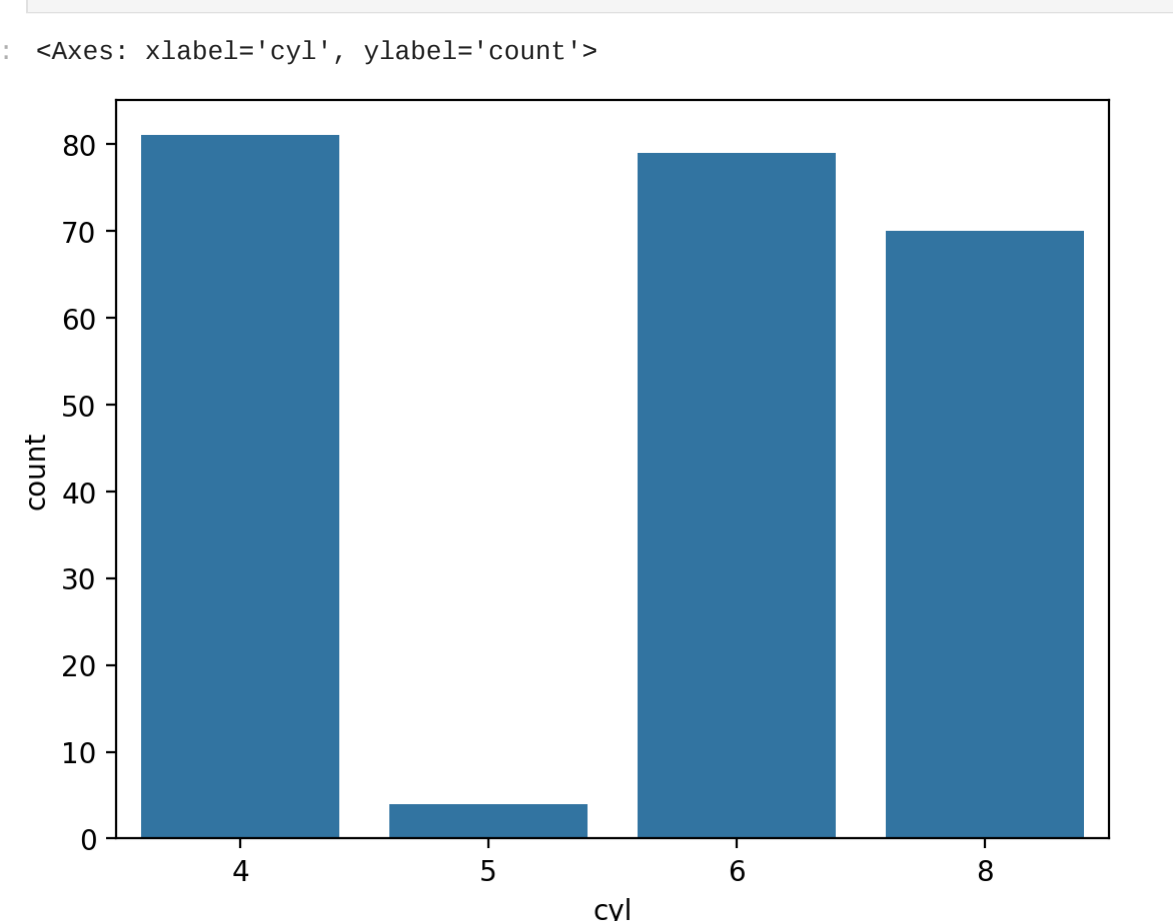
Out[8]: <Axes: xlabel='class', ylabel='count'>



count discrete variable

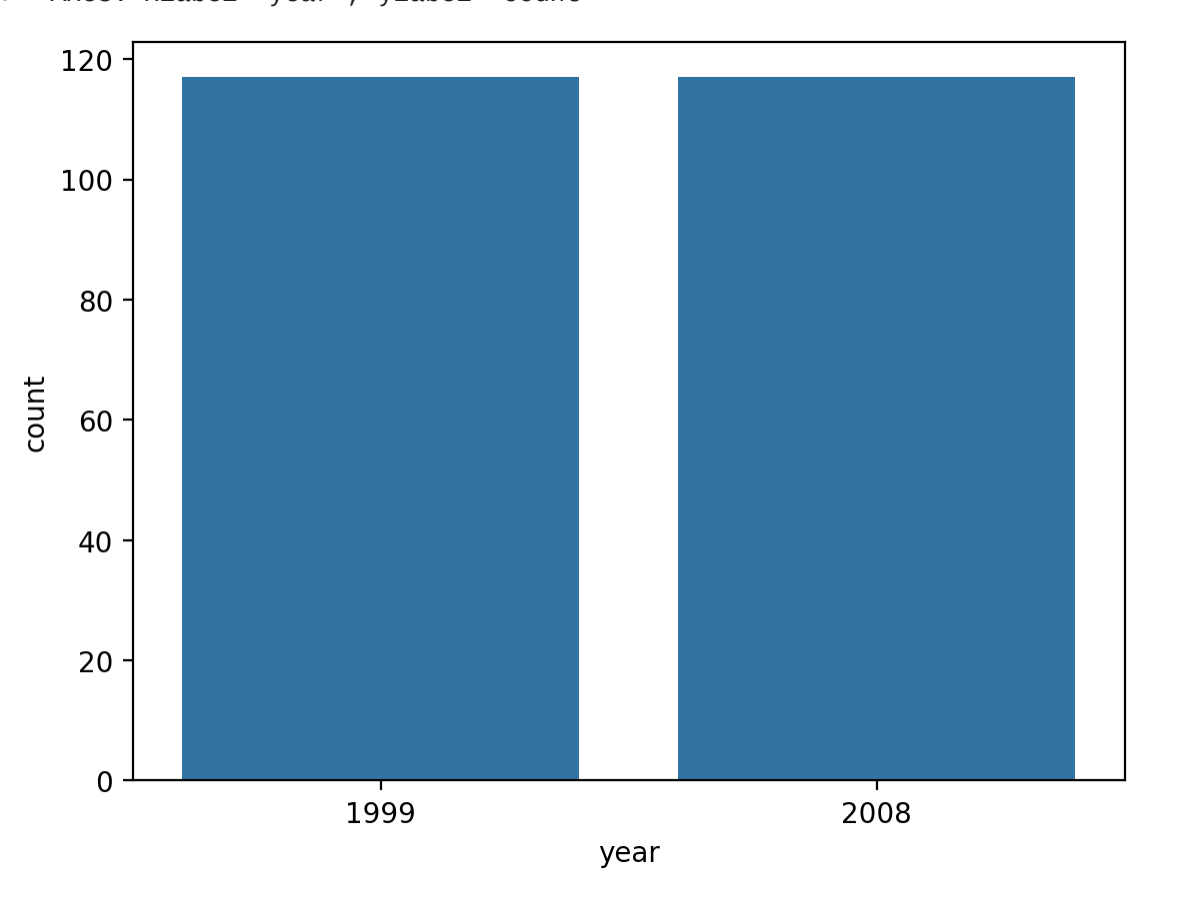
In [9]:
```
sns.countplot(x='cyl', data=df)
```

Out[9]: <Axes: xlabel='cyl', ylabel='count'>



In [10]:
```
sns.countplot(x='year', data=df)
```

Out[10]: <Axes: xlabel='year', ylabel='count'>



countplot and arrange value

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

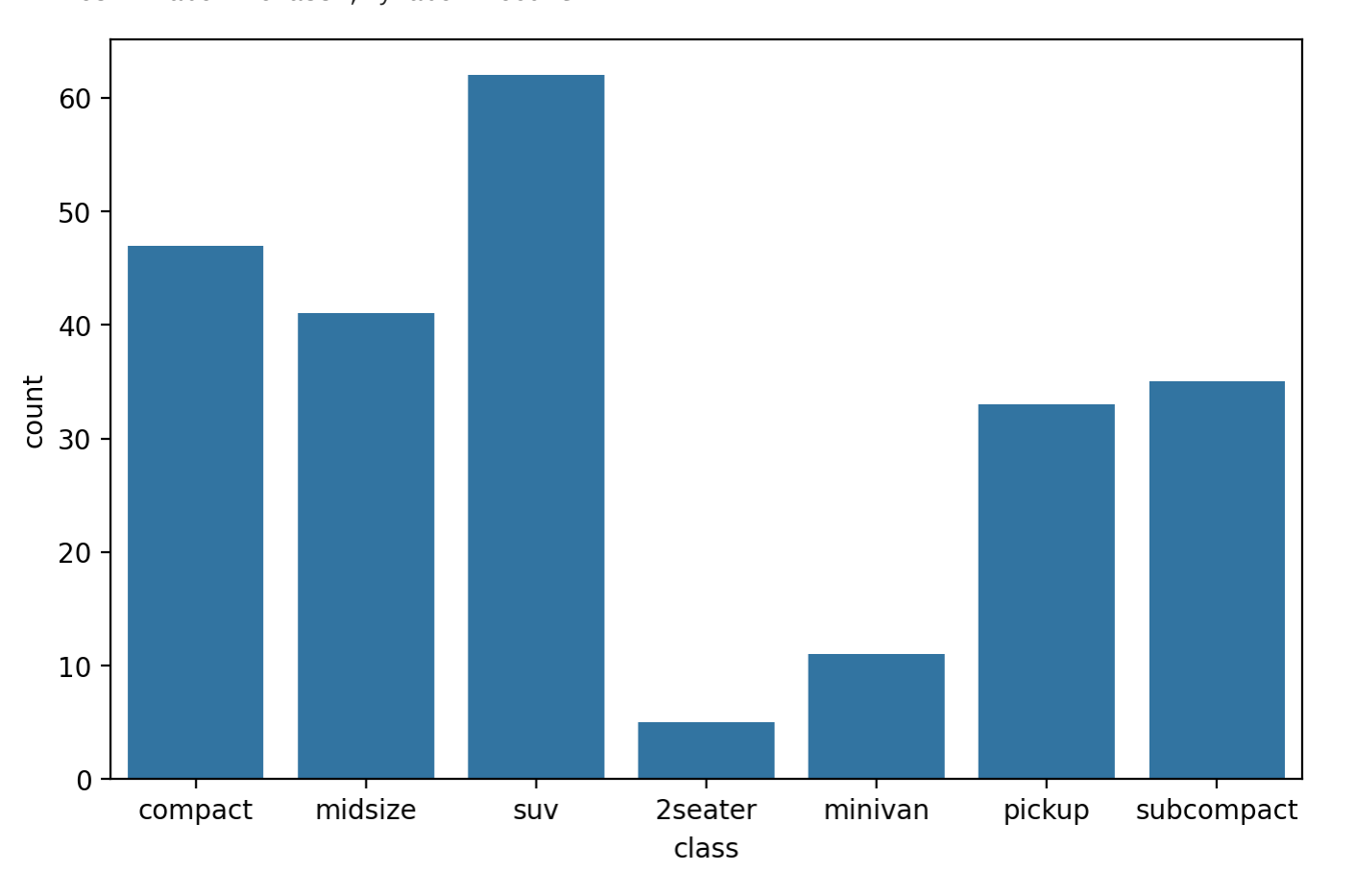
Out[11]:
```
class
suv           62
compact       47
midsize       41
subcompact    35
pickup        33
minivan       11
2seater        5
Name: count, dtype: int64
```

In [12]:
```
df['class'].value_counts().index
```

Out[12]:
```
Index(['suv', 'compact', 'midsize', 'subcompact', 'pickup', 'minivan',
       '2seater'],
      dtype='object', name='class')
```

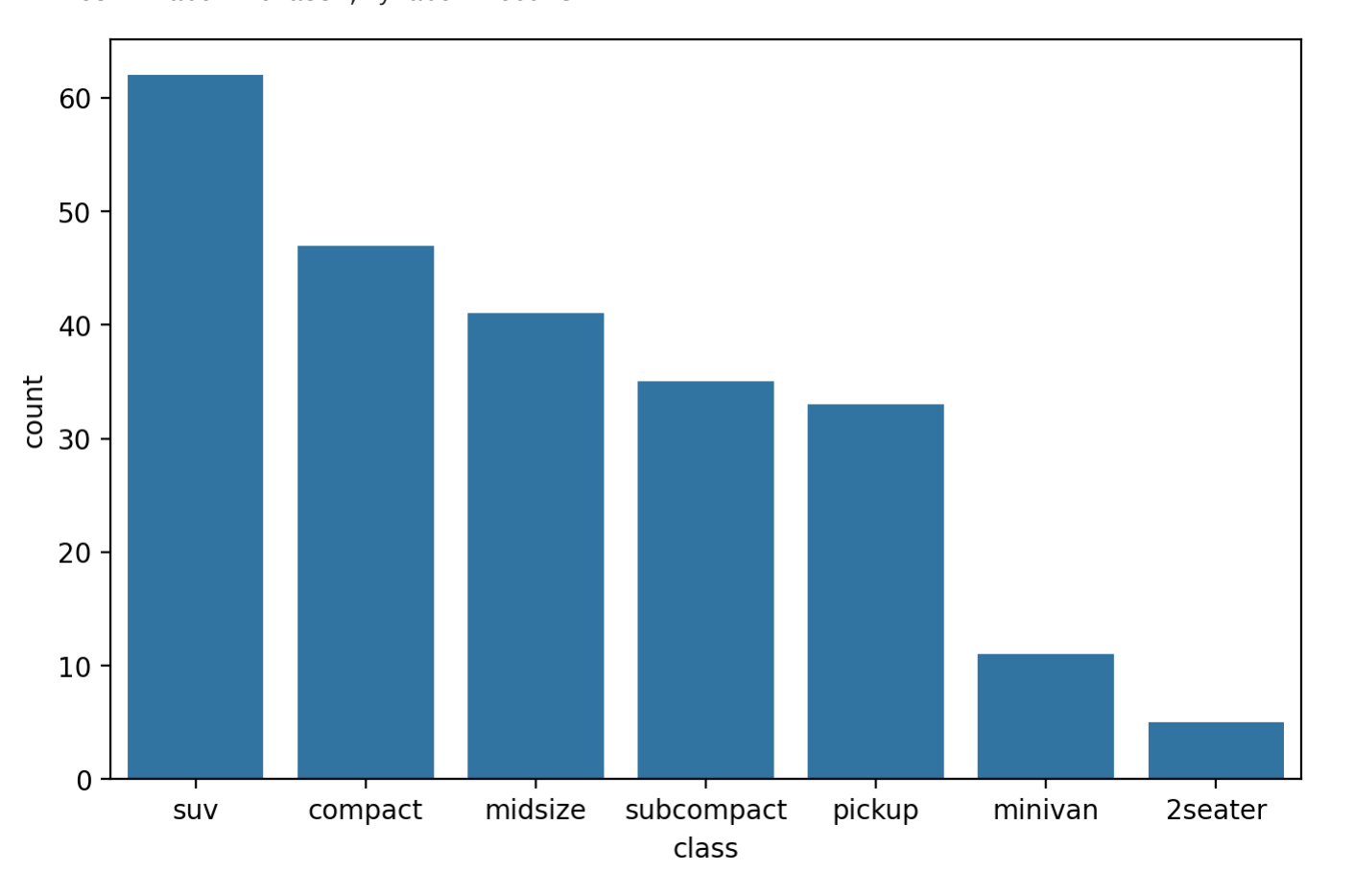
In [13]:
```
plt.figure(figsize=(8,5))
sns.countplot(x='class', data = df)
```

Out[13]: <Axes: xlabel='class', ylabel='count'>



In [14]:
```
plt.figure(figsize=(8,5))
sns.countplot(x='class', data = df,
              order=df['class'].value_counts().index)
```

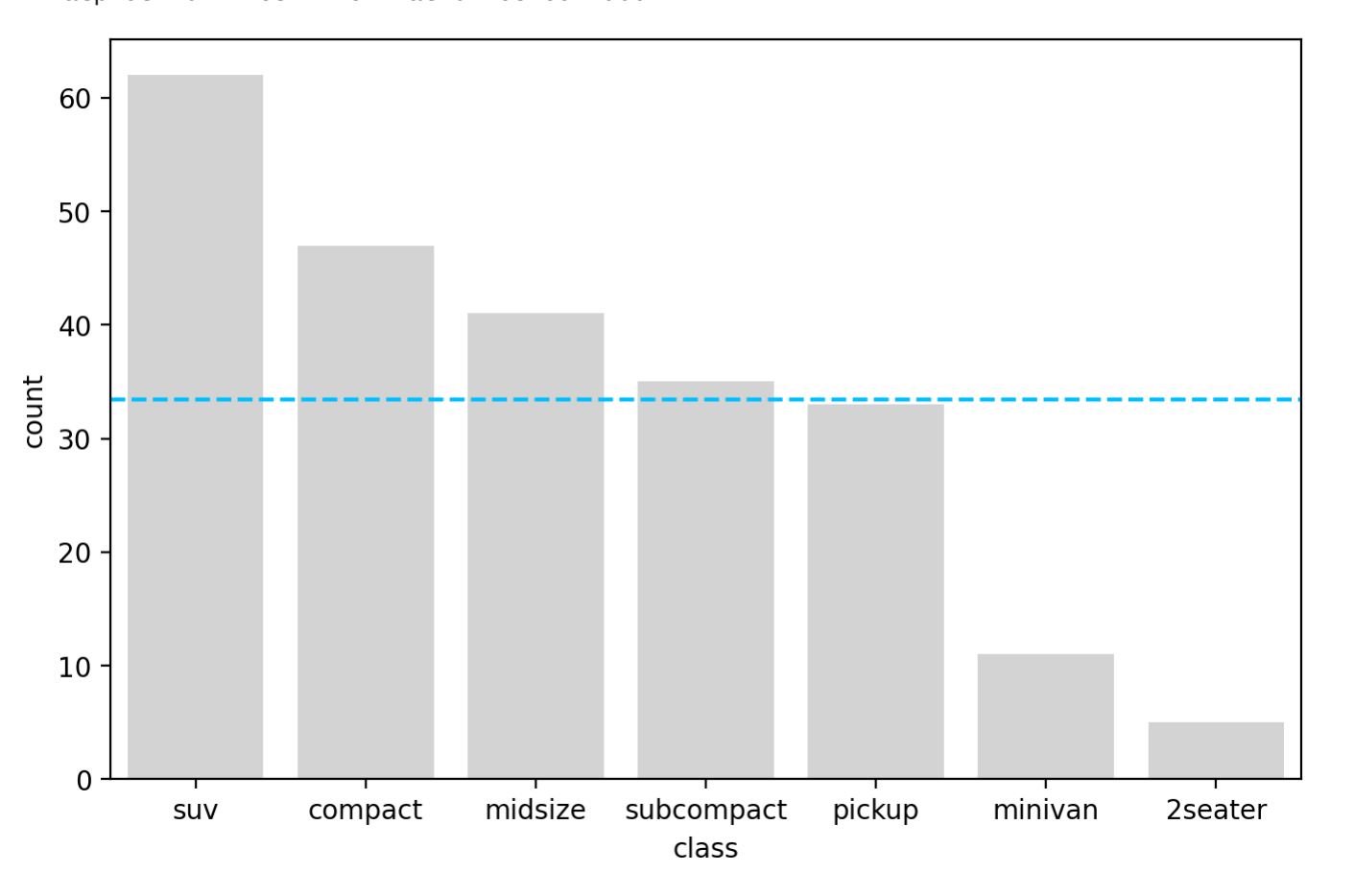
Out[14]: <Axes: xlabel='class', ylabel='count'>



add horizontal line

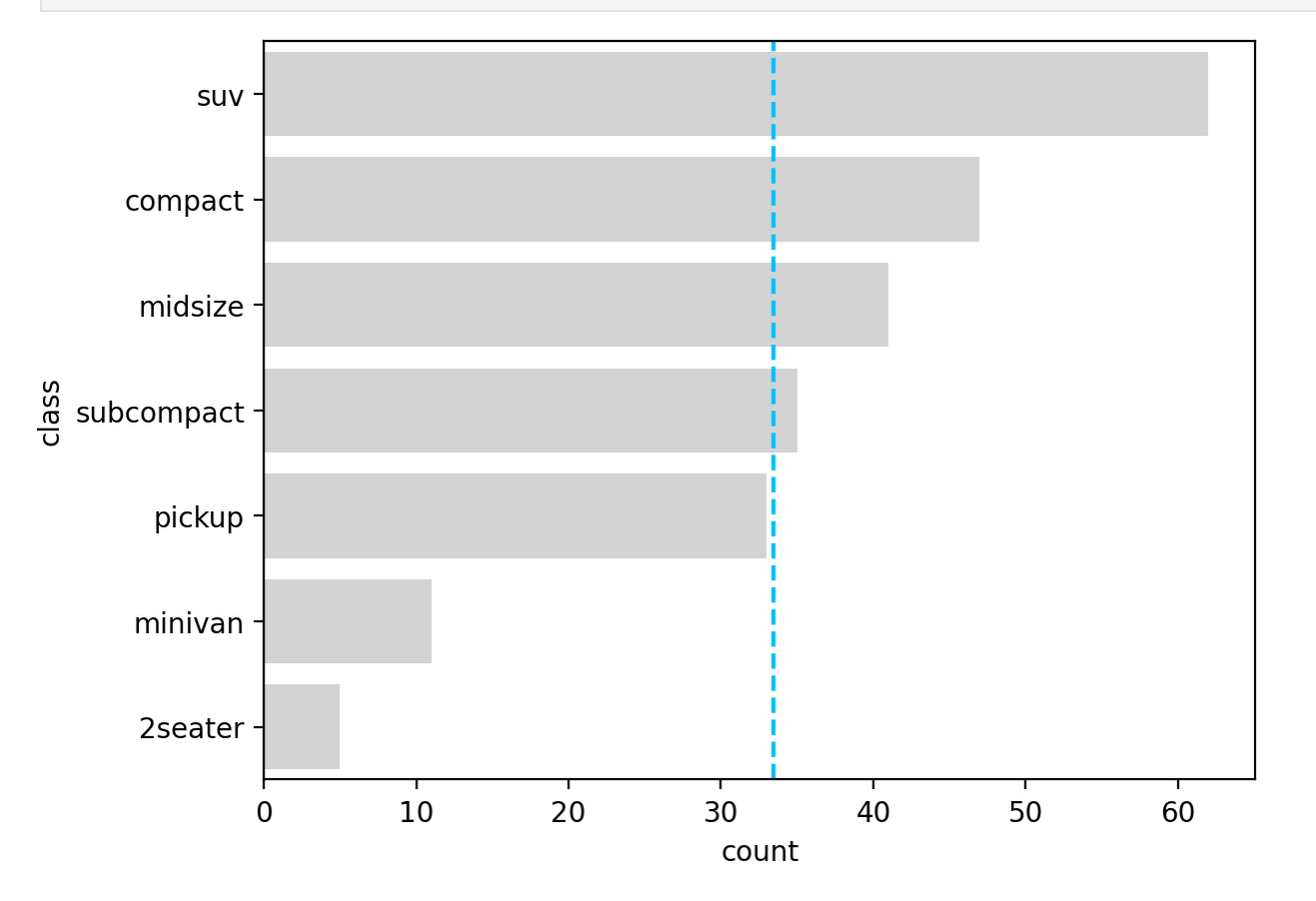
In [15]:
```
plt.figure(figsize=(8,5))
avg = df['class'].value_counts().mean()

sns.countplot(x = 'class',
             data = df,
             color = 'lightgray',
             order = df['class'].value_counts().index)
plt.axhline(y = avg, linestyle = '--', color = 'deepskyblue') # axhline horizontal
```

Out[15]: <matplotlib.lines.Line2D at 0x1c32c91fd00>



add vertical line

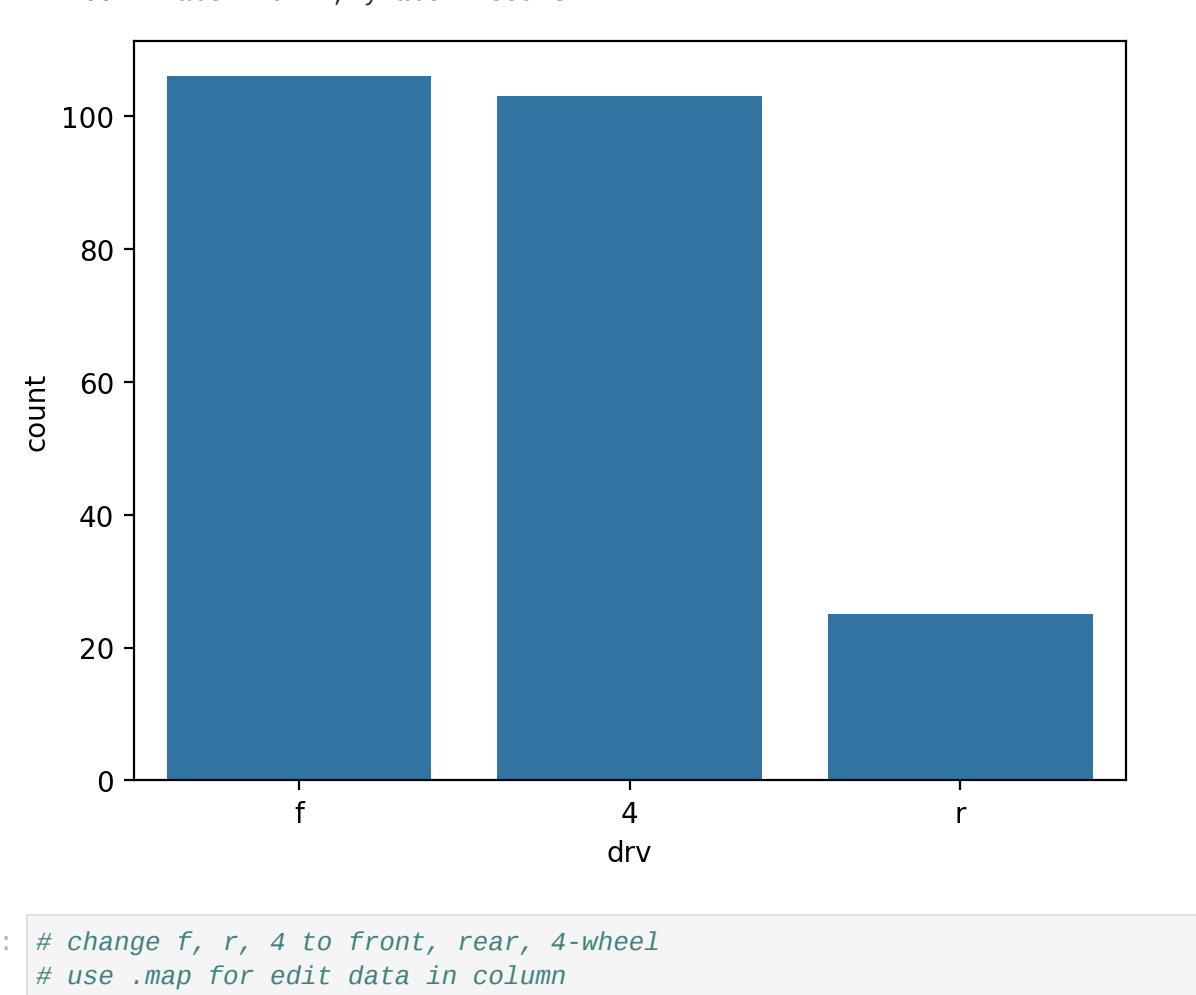
In [16]:
```
avg = df['class'].value_counts().mean()

sns.countplot(y = 'class',
             data = df,
             color = 'lightgray',
             order = df['class'].value_counts().index)
plt.axvline(x = avg, linestyle = '--', color = 'deepskyblue') # axvline vertical
plt.ylabel = ''
```



categorize column

In [20]:
```
# before edit
sns.countplot(x = 'drv', data = df)
```

Out[20]: <Axes: xlabel='drv', ylabel='count'>



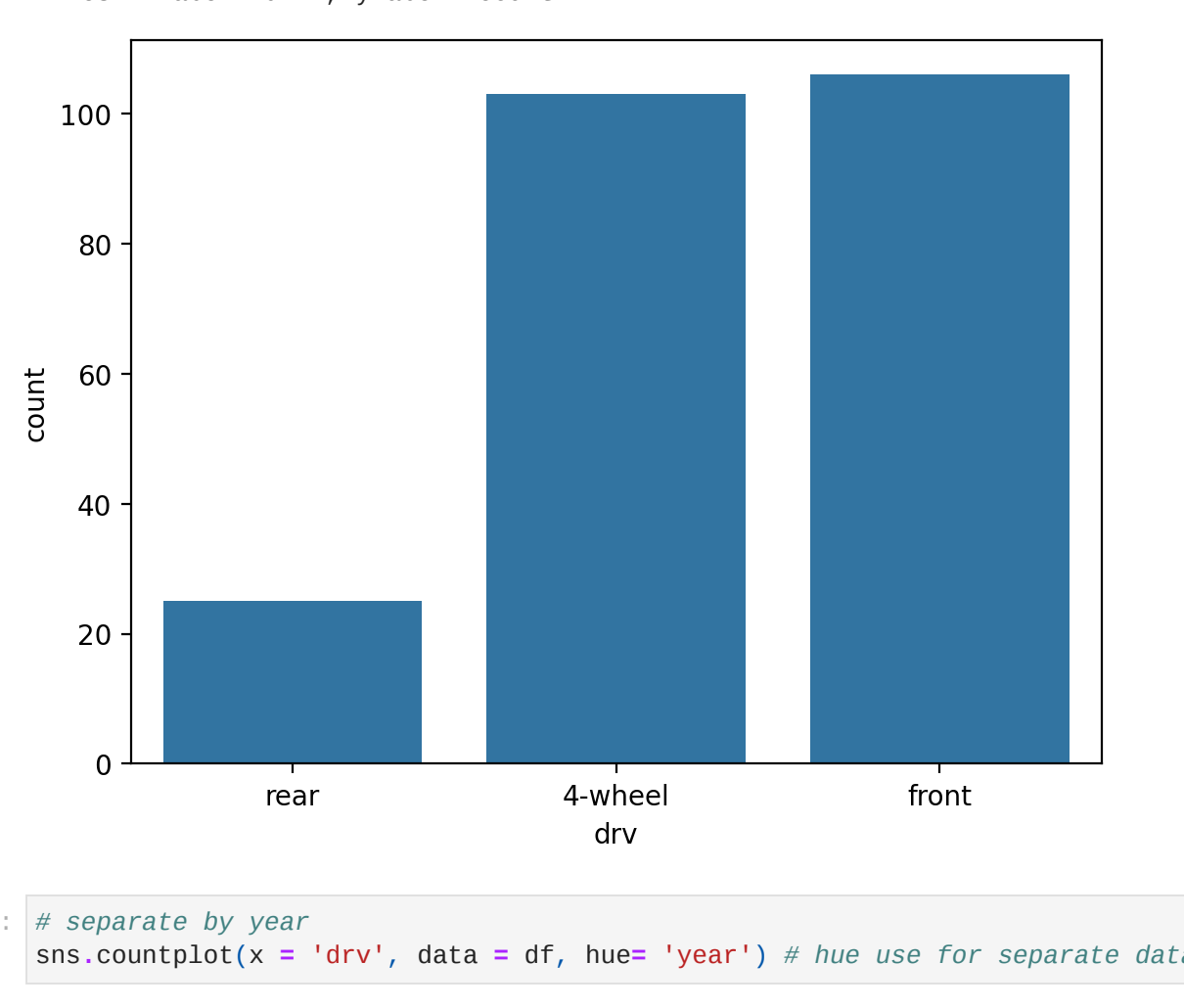
In [21]:
```
# change f, r, 4 to front, rear, 4-wheel
# use .map for edit data in column
df['drv'] = df.drv.map({'f':'front', 'r':'rear', '4':'4-wheel'}).astype('category')
df.head()
```

Out[21]:

	manufacturer	model	displ	year	cyl	trans	drv	cty	hwy	fl	class
0	audi	a4	1.8	1999	4	auto(l5)	front	18	29	p	compact
1	audi	a4	1.8	1999	4	manual(m5)	front	21	29	p	compact
2	audi	a4	2.0	2008	4	manual(m6)	front	20	31	p	compact
3	audi	a4	2.0	2008	4	auto(av)	front	21	30	p	compact
4	audi	a4	2.8	1999	6	auto(l5)	front	16	26	p	compact

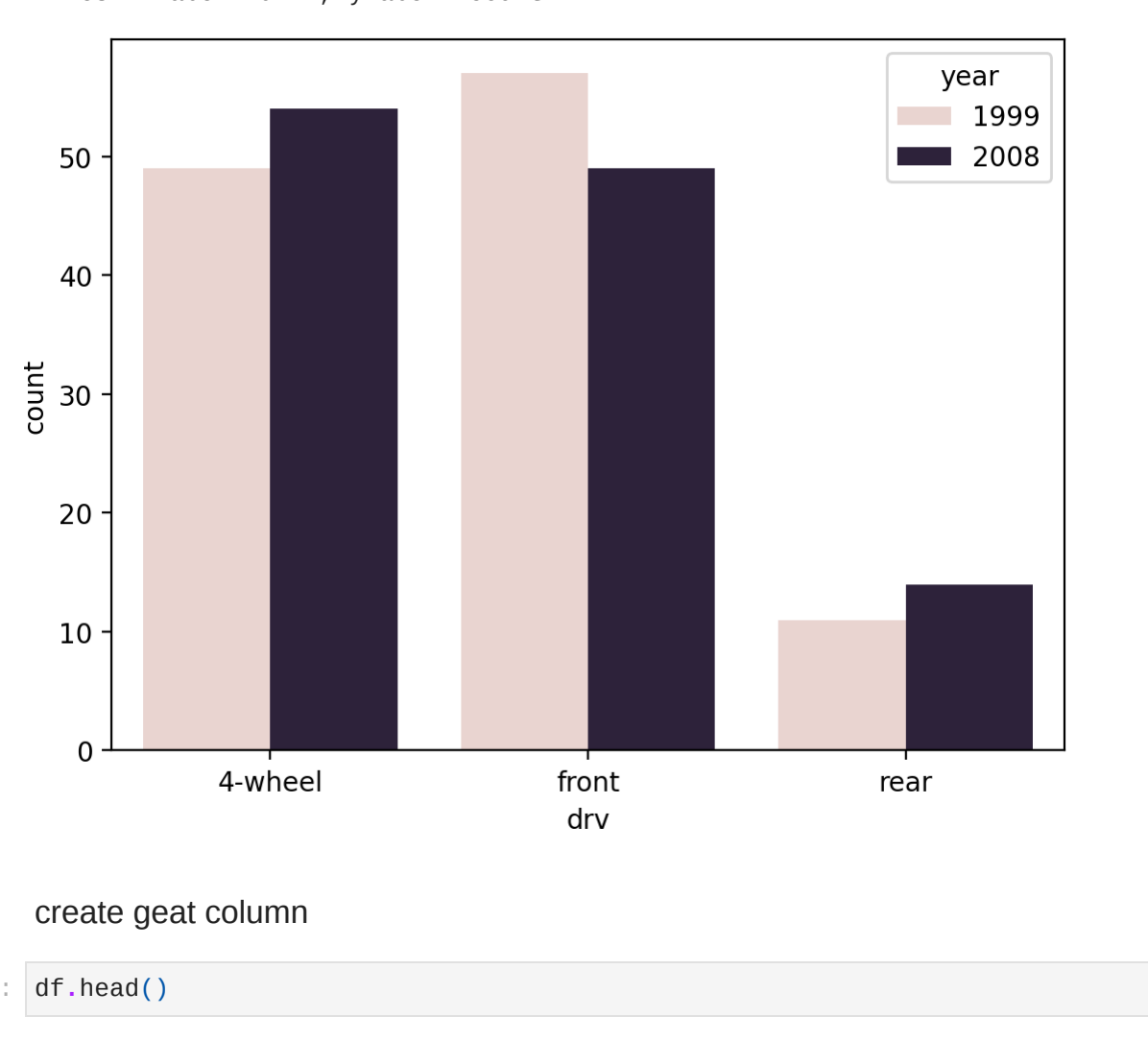
In [22]:
```
# after edit
# use order for arrange
sns.countplot(x = 'drv', data = df, order = ['rear', '4-wheel', 'front'])
```

Out[22]: <Axes: xlabel='drv', ylabel='count'>



In [23]:
```
# separate by year
sns.countplot(x = 'drv', data = df, hue= 'year') # hue use for separate data by other column
```

Out[23]: <Axes: xlabel='drv', ylabel='count'>



create geat column

In [24]:
```
df.head()
```

Out[24]:

	manufacturer	model	displ	year	cyl	trans	drv	cty	hwy	fl	class
0	audi	a4	1.8	1999	4	auto(l5)	front	18	29	p	compact
1	audi	a4	1.8	1999	4	manual(m5)	front	21	29	p	compact
2	audi	a4	2.0	2008	4	manual(m6)	front	20	31	p	compact
3	audi	a4	2.0	2008	4	auto(av)	front	21	30	p	compact
4	audi	a4	2.8	1999	6	auto(l5)	front	16	26	p	compact

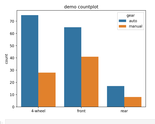
In [25]:
```
df['gear'] = df.trans.str[0].map({'a':'auto','m':'manual' }).astype('category')
df.head()
```

Out[25]:

	manufacturer	model	displ	year	cyl	trans	drv	cty	hwy	fl	class	gear
0	audi	a4	1.8	1999	4	auto(l5)	front	18	29	p	compact	auto
1	audi	a4	1.8	1999	4	manual(m5)	front	21	29	p	compact	manual
2	audi	a4	2.0	2008	4	manual(m6)	front	20	31	p	compact	manual
3	audi	a4	2.0	2008	4	auto(av)	front	21	30	p	compact	auto
4	audi	a4	2.8	1999	6	auto(l5)	front	16	26	p	compact	auto

In [26]:
```
sns.countplot(x = 'drv', data = df, hue='gear' )
plt.xlabel('') # for not show name of x axis
plt.title('demo countplot')
```

Out[26]: Text(0.5, 1.0, 'demo countplot')


demo countplot
gear
auto
manual
count
70
60
50
40
30
20
10
0
4-wheel
front
rear

In [ ]: