Assignment - 2

Name : Vajjah Srinivasa Taaran

Reg no : 21BCE8745

In [1]:

```
import matplotlib.pyplot as plt
import seaborn as sns
```

In [2]:

```
sns.get_dataset_names()
```

Out[2]:

```
['anagrams',
 'anscombe',
 'attention',
 'brain_networks',
 'car_crashes',
 'diamonds',
 'dots',
 'dowjones',
 'exercise',
 'flights',
 'fmri',
 'geyser',
 'glue',
 'healthexp',
 'iris',
 'mpg',
 'penguins',
 'planets',
 'seaice',
 'taxis',
 'tips',
 'titanic']
```

In [3]:

```
df = sns.load_dataset('car_crashes')
```

In [4]:

```
df
```

Out[4]:

	total	speeding	alcohol	not_distracted	no_previous	ins_premium	ins_losses	abbrev
0	18.8	7.332	5.640	18.048	15.040	784.55	145.08	AL
1	18.1	7.421	4.525	16.290	17.014	1053.48	133.93	AK
2	18.6	6.510	5.208	15.624	17.856	899.47	110.35	AZ
3	22.4	4.032	5.824	21.056	21.280	827.34	142.39	AR
4	12.0	4.200	3.360	10.920	10.680	878.41	165.63	CA
5	13.6	5.032	3.808	10.744	12.920	835.50	139.91	CO
6	10.8	4.968	3.888	9.396	8.856	1068.73	167.02	CT
7	16.2	6.156	4.860	14.094	16.038	1137.87	151.48	DE
8	5.9	2.006	1.593	5.900	5.900	1273.89	136.05	DC
9	17.9	3.759	5.191	16.468	16.826	1160.13	144.18	FL
10	15.6	2.964	3.900	14.820	14.508	913.15	142.80	GA
11	17.5	9.450	7.175	14.350	15.225	861.18	120.92	HI
12	15.3	5.508	4.437	13.005	14.994	641.96	82.75	ID
13	12.8	4.608	4.352	12.032	12.288	803.11	139.15	IL
14	14.5	3.625	4.205	13.775	13.775	710.46	108.92	IN
15	15.7	2.669	3.925	15.229	13.659	649.06	114.47	IA
16	17.8	4.806	4.272	13.706	15.130	780.45	133.80	KS
17	21.4	4.066	4.922	16.692	16.264	872.51	137.13	KY
18	20.5	7.175	6.765	14.965	20.090	1281.55	194.78	LA
19	15.1	5.738	4.530	13.137	12.684	661.88	96.57	ME
20	12.5	4.250	4.000	8.875	12.375	1048.78	192.70	MD
21	8.2	1.886	2.870	7.134	6.560	1011.14	135.63	MA
22	14.1	3.384	3.948	13.395	10.857	1110.61	152.26	MI
23	9.6	2.208	2.784	8.448	8.448	777.18	133.35	MN
24	17.6	2.640	5.456	1.760	17.600	896.07	155.77	MS
25	16.1	6.923	5.474	14.812	13.524	790.32	144.45	MO
26	21.4	8.346	9.416	17.976	18.190	816.21	85.15	MT
27	14.9	1.937	5.215	13.857	13.410	732.28	114.82	NE
28	14.7	5.439	4.704	13.965	14.553	1029.87	138.71	NV
29	11.6	4.060	3.480	10.092	9.628	746.54	120.21	NH
30	11.2	1.792	3.136	9.632	8.736	1301.52	159.85	NJ
31	18.4	3.496	4.968	12.328	18.032	869.85	120.75	NM
32	12.3	3.936	3.567	10.824	9.840	1234.31	150.01	NY
33	16.8	6.552	5.208	15.792	13.608	708.24	127.82	NC
34	23.9	5.497	10.038	23.661	20.554	688.75	109.72	ND
35	14.1	3.948	4.794	13.959	11.562	697.73	133.52	OH

	total	speeding	alcohol	not_distracted	no_previous	ins_premium	ins_losses	abbrev
36	19.9	6.368	5.771	18.308	18.706	881.51	178.86	OK
37	12.8	4.224	3.328	8.576	11.520	804.71	104.61	OR
38	18.2	9.100	5.642	17.472	16.016	905.99	153.86	PA
39	11.1	3.774	4.218	10.212	8.769	1148.99	148.58	RI
40	23.9	9.082	9.799	22.944	19.359	858.97	116.29	SC
41	19.4	6.014	6.402	19.012	16.684	669.31	96.87	SD
42	19.5	4.095	5.655	15.990	15.795	767.91	155.57	TN
43	19.4	7.760	7.372	17.654	16.878	1004.75	156.83	TX
44	11.3	4.859	1.808	9.944	10.848	809.38	109.48	UT
45	13.6	4.080	4.080	13.056	12.920	716.20	109.61	VT
46	12.7	2.413	3.429	11.049	11.176	768.95	153.72	VA
47	10.6	4.452	3.498	8.692	9.116	890.03	111.62	WA
48	23.8	8.092	6.664	23.086	20.706	992.61	152.56	WV
49	13.8	4.968	4.554	5.382	11.592	670.31	106.62	WI
50	17.4	7.308	5.568	14.094	15.660	791.14	122.04	WY

In [5]:

```
df.head()
```

Out[5]:

	total	speeding	alcohol	not_distracted	no_previous	ins_premium	ins_losses	abbrev
0	18.8	7.332	5.640	18.048	15.040	784.55	145.08	AL
1	18.1	7.421	4.525	16.290	17.014	1053.48	133.93	AK
2	18.6	6.510	5.208	15.624	17.856	899.47	110.35	AZ
3	22.4	4.032	5.824	21.056	21.280	827.34	142.39	AR
4	12.0	4.200	3.360	10.920	10.680	878.41	165.63	CA

In [6]:

```
df.tail()
```

Out[6]:

	total	speeding	alcohol	not_distracted	no_previous	ins_premium	ins_losses	abbrev
46	12.7	2.413	3.429	11.049	11.176	768.95	153.72	VA
47	10.6	4.452	3.498	8.692	9.116	890.03	111.62	WA
48	23.8	8.092	6.664	23.086	20.706	992.61	152.56	WV
49	13.8	4.968	4.554	5.382	11.592	670.31	106.62	WI
50	17.4	7.308	5.568	14.094	15.660	791.14	122.04	WY

In [7]:

```
df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 51 entries, 0 to 50
Data columns (total 8 columns):
 #   Column          Non-Null Count  Dtype  
---  ------          --------------  -----  
 0   total           51 non-null     float64
 1   speeding        51 non-null     float64
 2   alcohol         51 non-null     float64
 3   not_distracted  51 non-null     float64
 4   no_previous     51 non-null     float64
 5   ins_premium     51 non-null     float64
 6   ins_losses      51 non-null     float64
 7   abbrev          51 non-null     object 
dtypes: float64(7), object(1)
memory usage: 3.3+ KB
```

In [8]:

```
df.shape
```

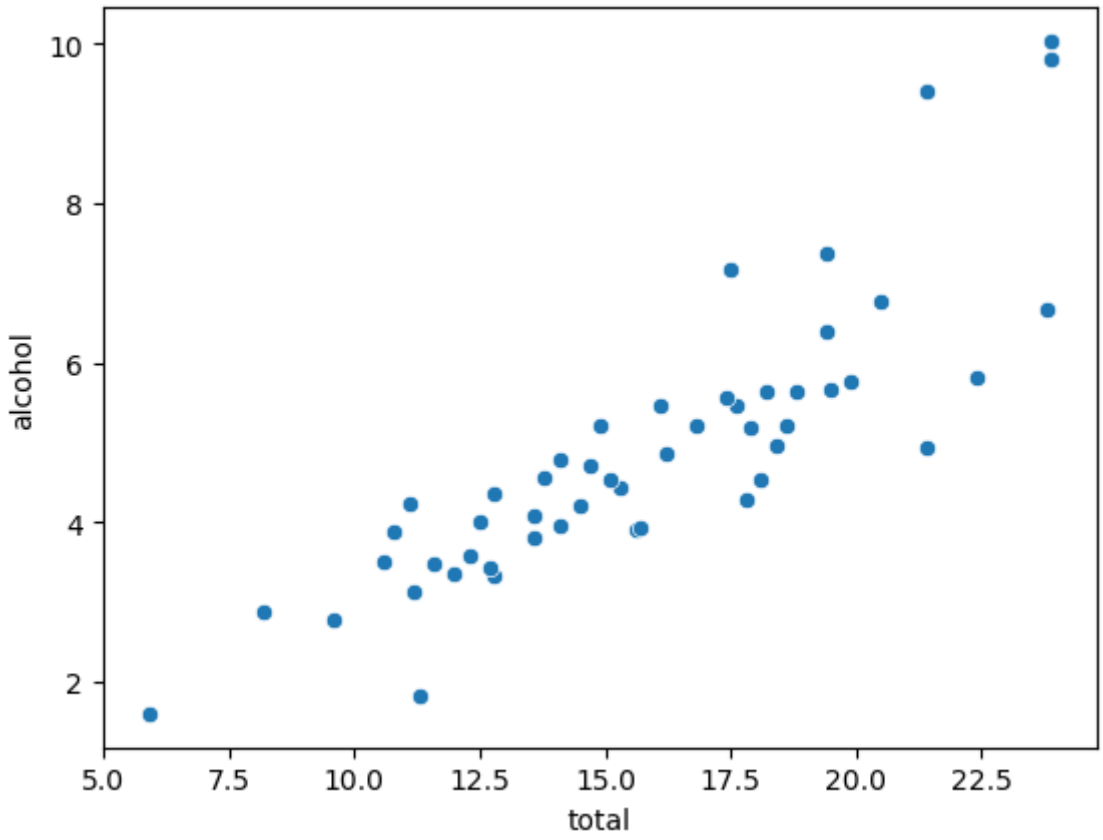
Out[8]:

```
(51, 8)
```

In [9]:

```
sns.scatterplot(x="total" , y ="alcohol",data = df)
print("Alcohol is directly proportional to total")
```

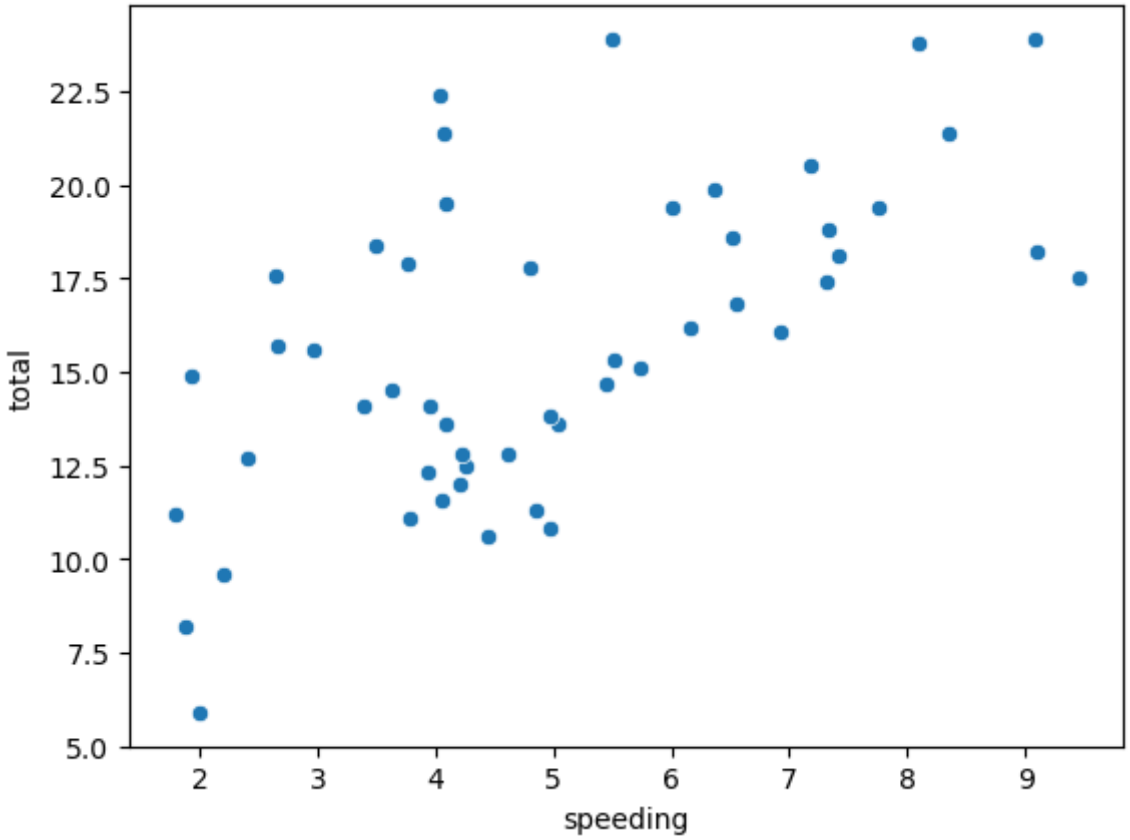
```
Alcohol is directly proportional to total
```



In [10]:

```
"speeding" , y ="total",data = df)
directly proportional to total whereas speeding is inversely proportional to total")
```

```
Alcohol is directly proportional to total whereas speeding is inversel
y proportional to total
```

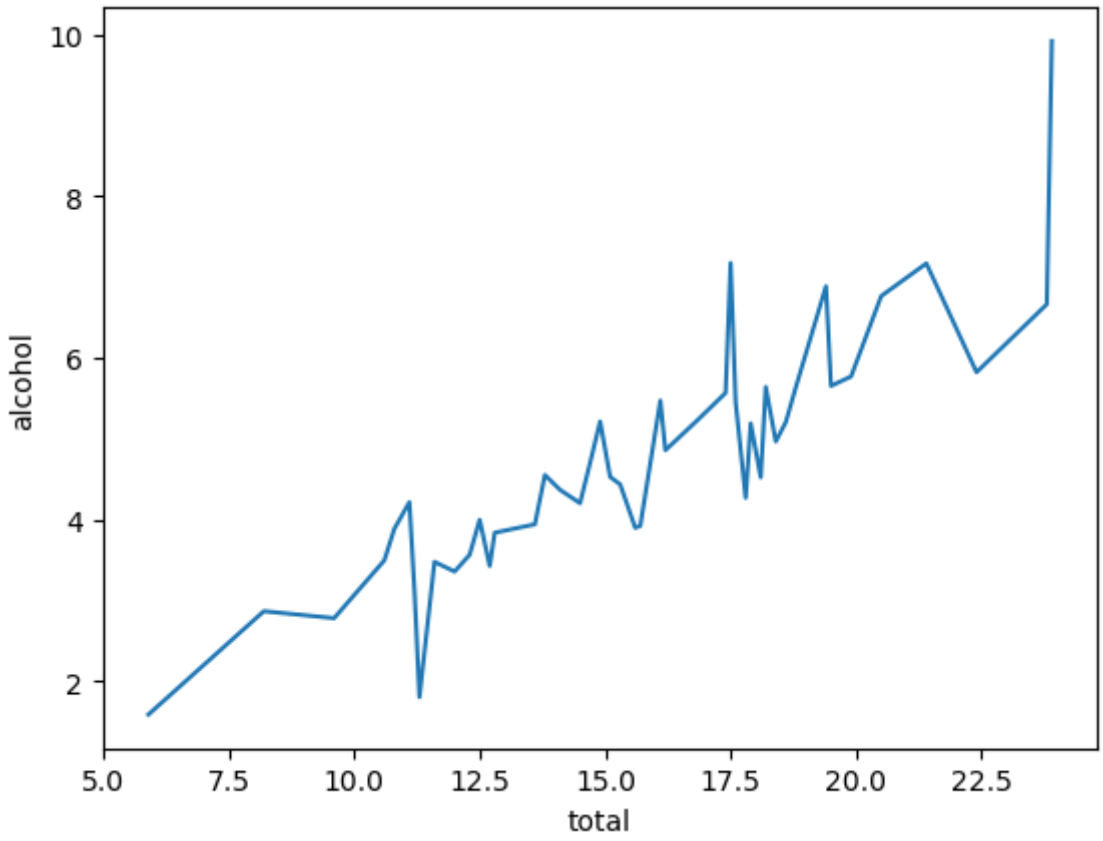


In [11]:

```
sns.lineplot(x="total" , y ="alcohol",data = df,errorbar= None)
```

Out[11]:

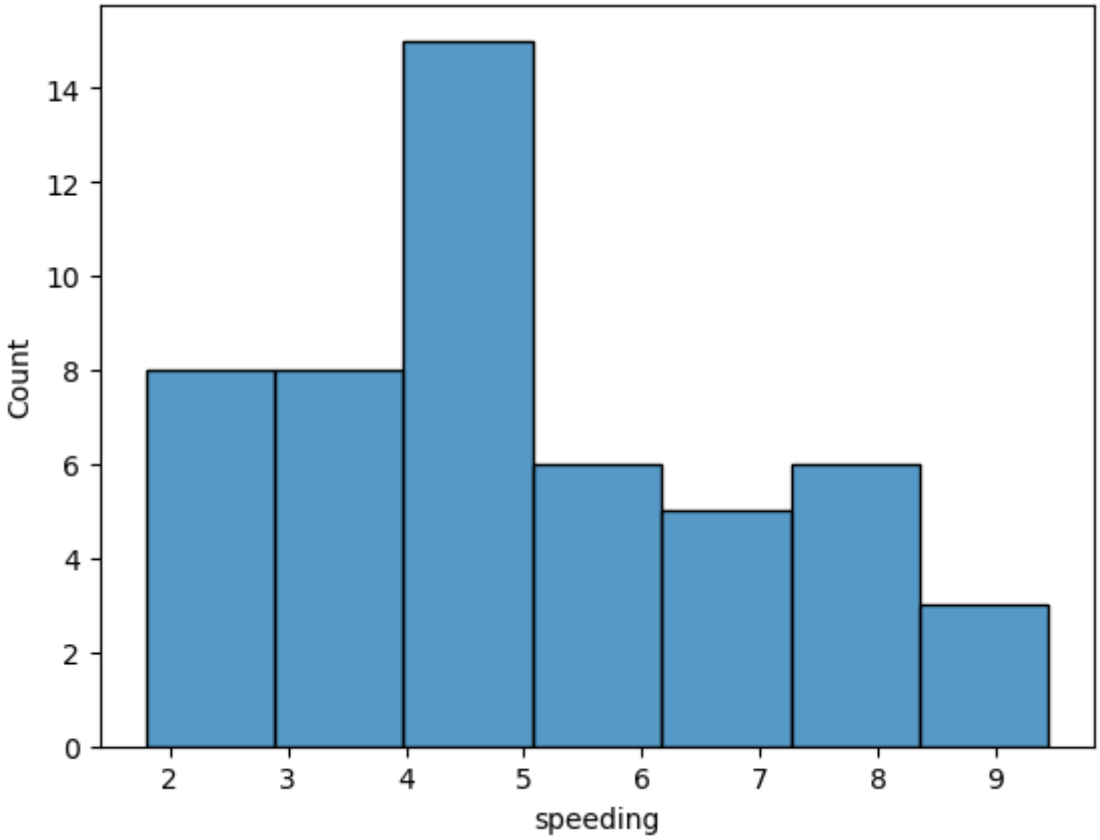
```
<Axes: xlabel='total', ylabel='alcohol'>
```



In [12]:

```
sns.histplot(df["speeding"])
print("Speeding vs Density distribution")
```

```
Speeding vs Density distribution
```



In [13]:

```
sns.relplot(x="speeding" , y ="alcohol",data =df,hue = "abbrev")
```

Out[13]:

```
<seaborn.axisgrid.FacetGrid at 0x14a4e7a90>
```

In [14]:

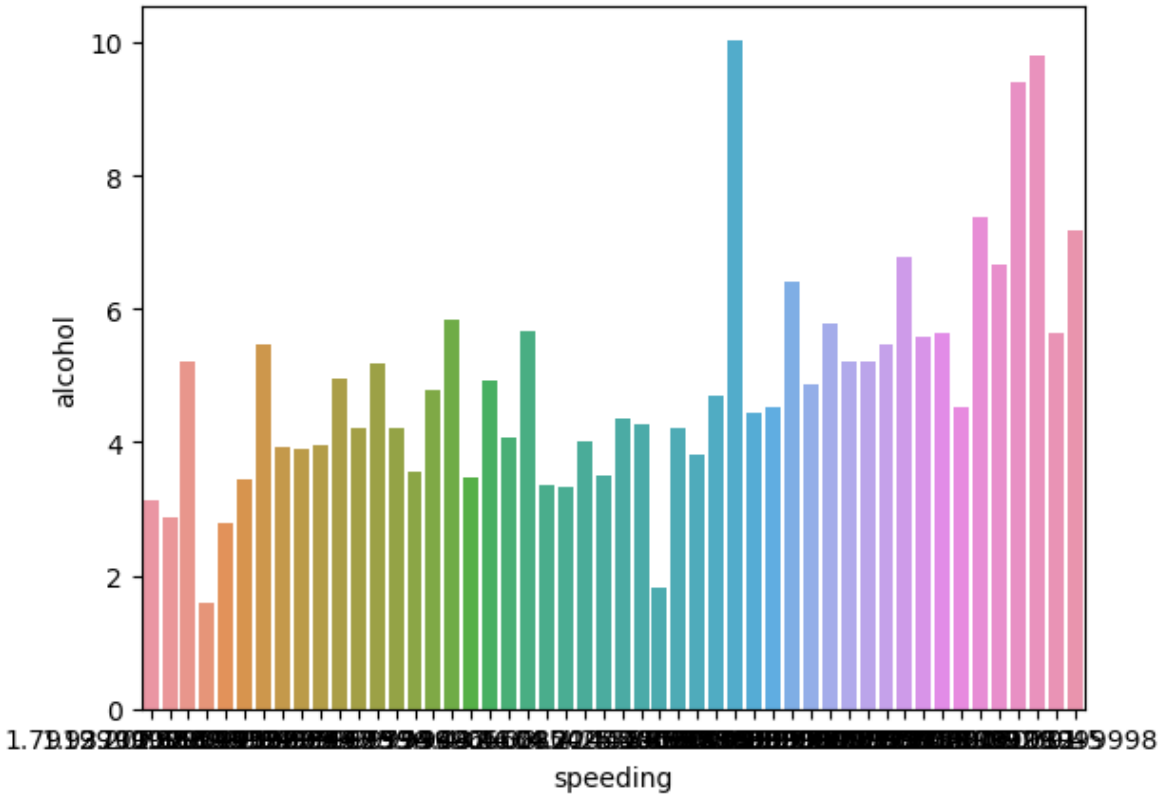
```
df["abbrev"].value_counts()
print("Here, we get count of all categories and has everything repeated")
```

```
Here, we get count of all categories and has everything repeated
```

In [15]:

```
sns.barplot(data = df,x = "speeding",y = "alcohol",errorbar = None)
print("bargraph: speeding vs alcohol")
```

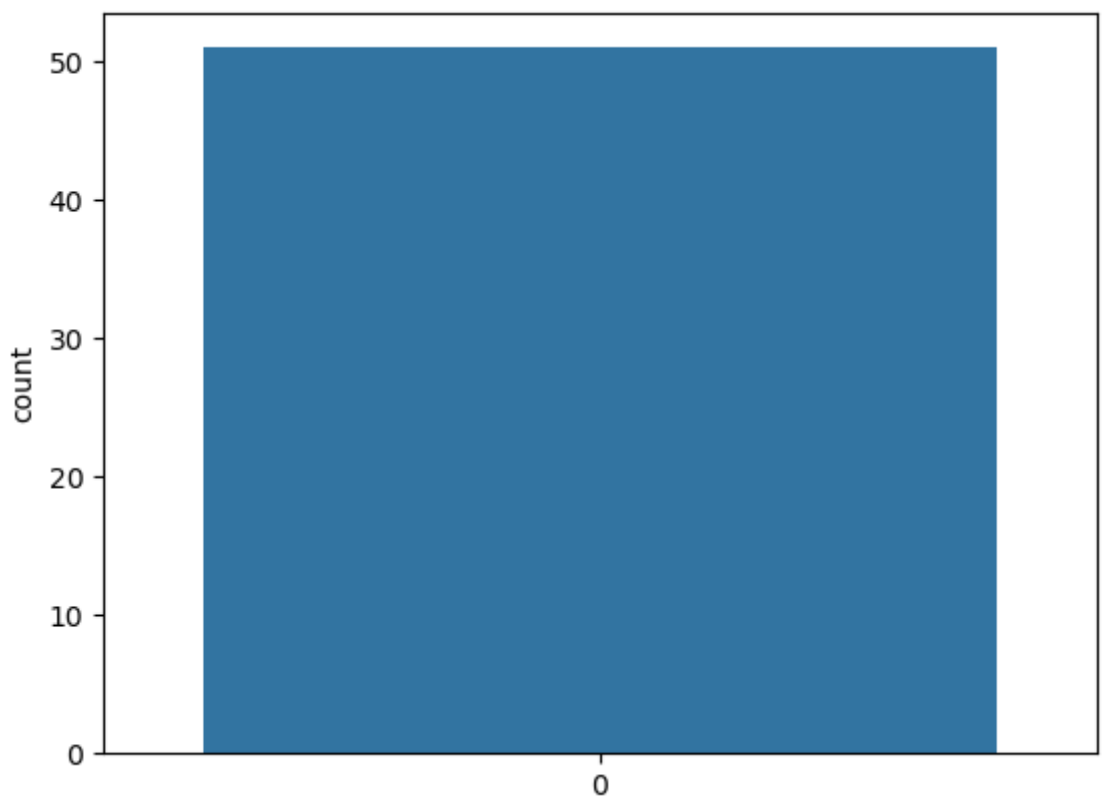
```
bargraph: speeding vs alcohol
```

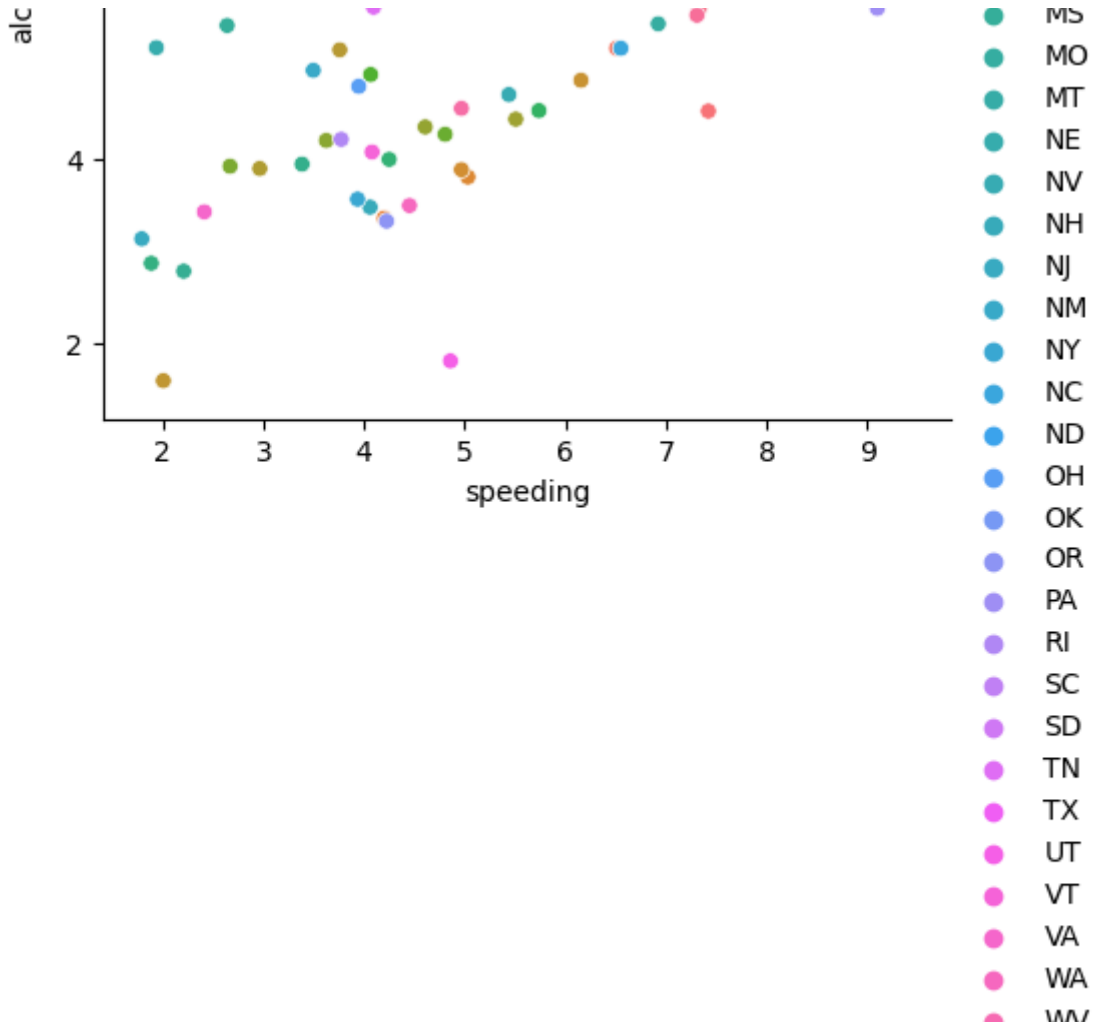


```
sns.countplot(df["total"])
```

Out[16]:

```
<Axes: ylabel='count'>
```



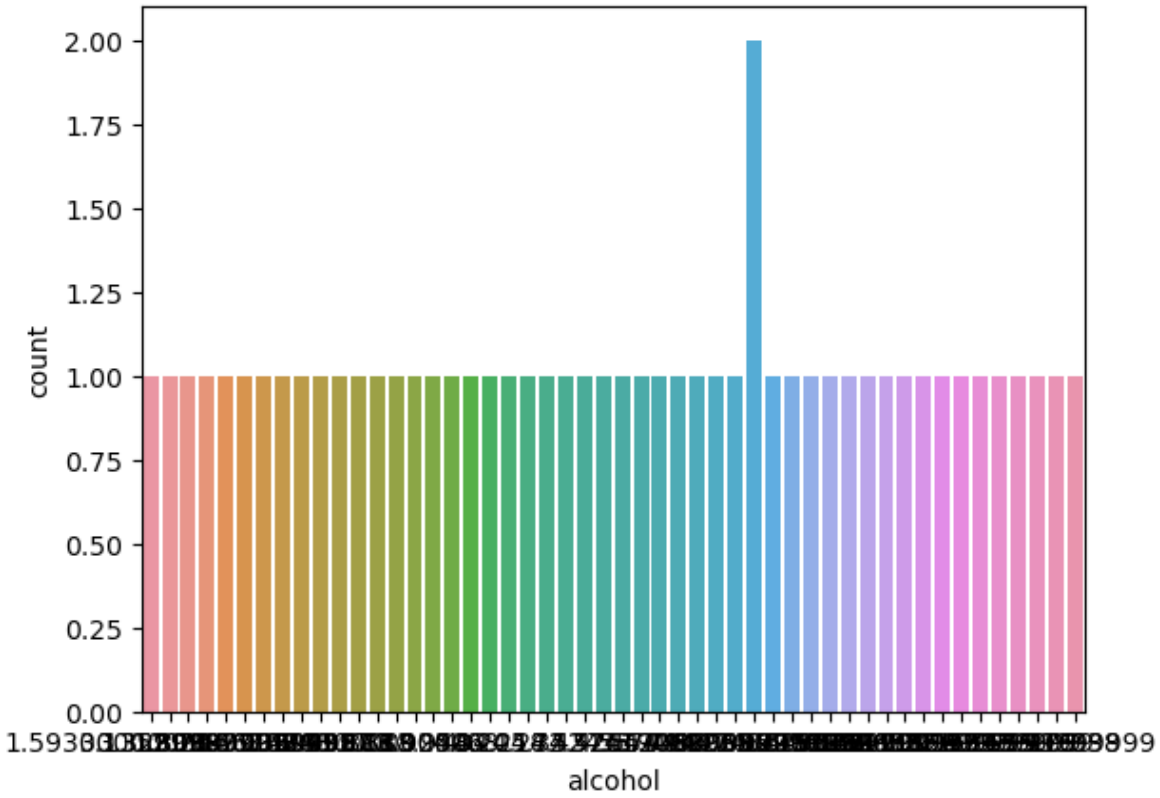


WV
WI
WY

```
sns.countplot(data= df,x = "alcohol")
```

Out[17]:

```
<Axes: xlabel='alcohol', ylabel='count'>
```

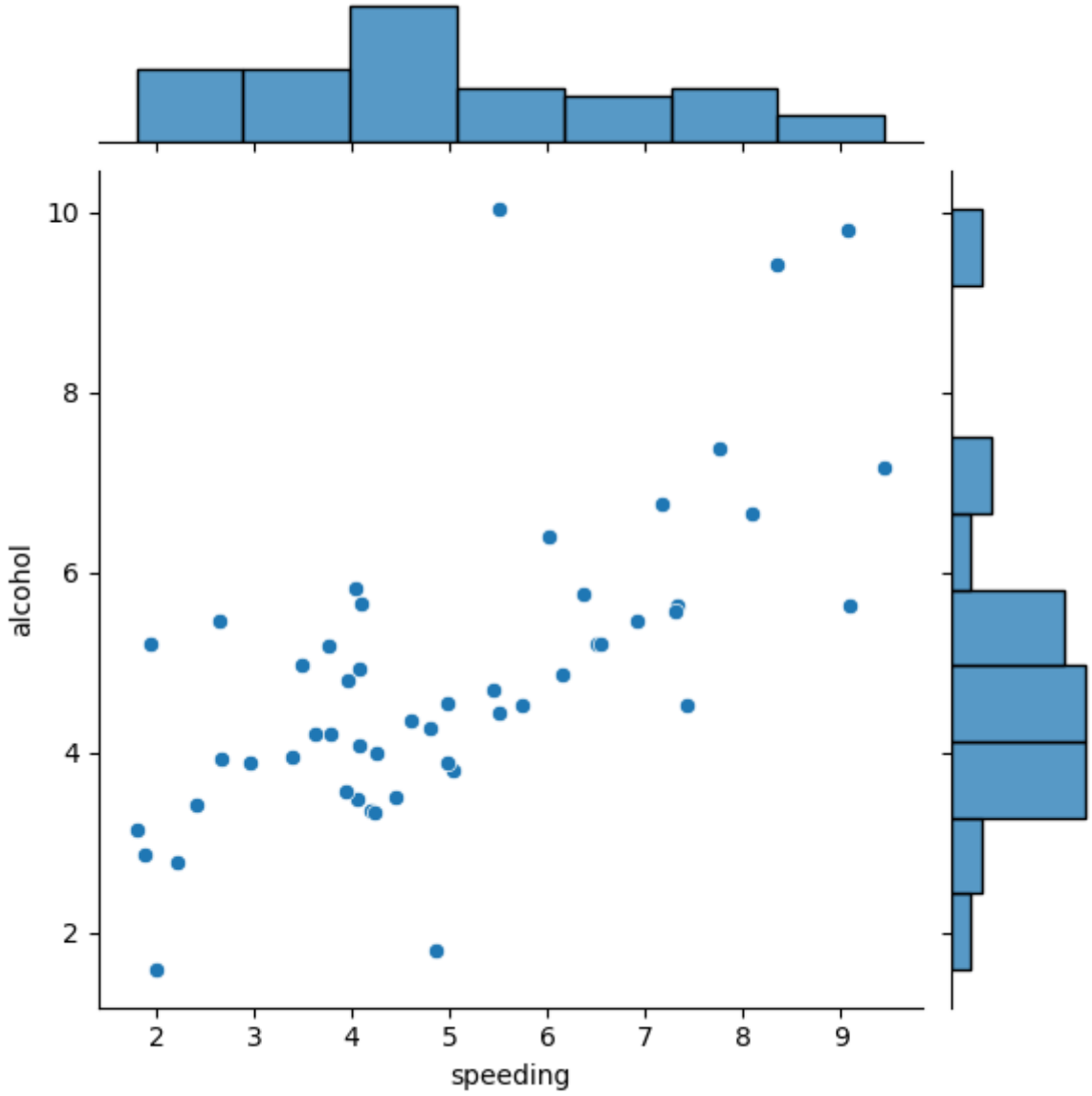


In [18]:

```
sns.jointplot(x="speeding" , y ="alcohol",data =df)
```

Out[18]:

```
<seaborn.axisgrid.JointGrid at 0x14a71ea10>
```

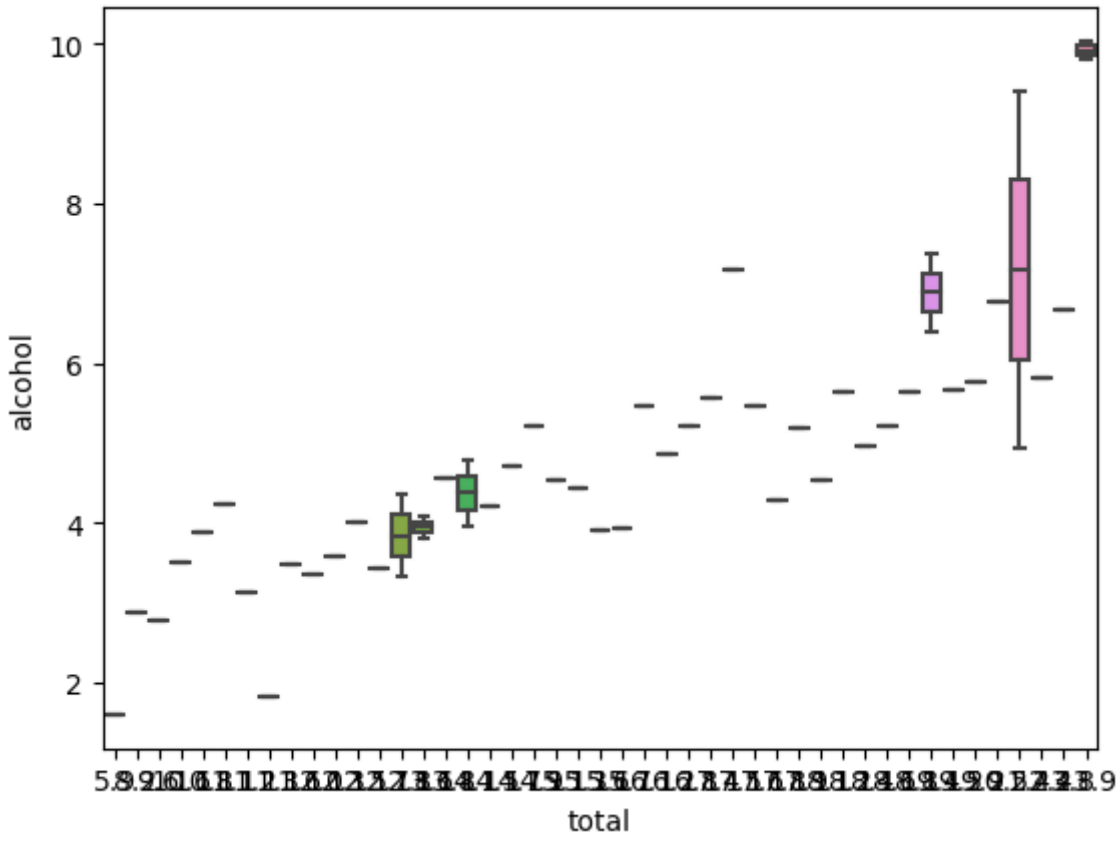


In [19]:

```
sns.boxplot(x="total" , y ="alcohol",data =df)
```

Out[19]:

```
<Axes: xlabel='total', ylabel='alcohol'>
```



In [20]:

```
corr= df.corr()
corr
```

```
/var/folders/0g/xqmh0yz92jx_s8ljsv3x08wr0000gn/T/ipykernel_36494/38645
96578.py:1: FutureWarning: The default value of numeric_only in DataFr
ame.corr is deprecated. In a future version, it will default to False.
Select only valid columns or specify the value of numeric_only to sile
nce this warning.
  corr= df.corr()
```

Out[20]:

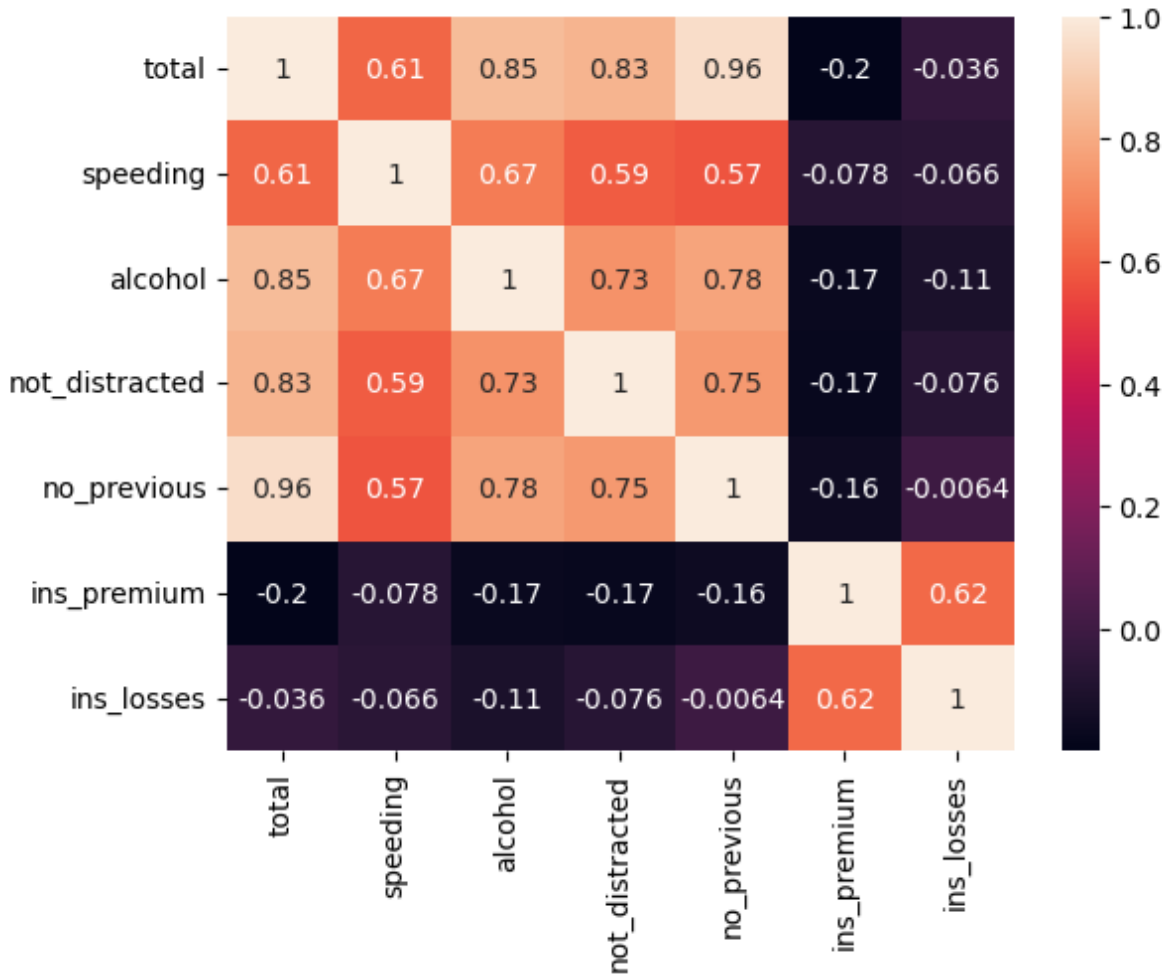
	total	speeding	alcohol	not_distracted	no_previous	ins_premium	ins_los
total	1.000000	0.611548	0.852613	0.827560	0.956179	-0.199702	-0.036
speeding	0.611548	1.000000	0.669719	0.588010	0.571976	-0.077675	-0.065
alcohol	0.852613	0.669719	1.000000	0.732816	0.783520	-0.170612	-0.112
not_distracted	0.827560	0.588010	0.732816	1.000000	0.747307	-0.174856	-0.075
no_previous	0.956179	0.571976	0.783520	0.747307	1.000000	-0.156895	-0.006
ins_premium	-0.199702	-0.077675	-0.170612	-0.174856	-0.156895	1.000000	0.623
ins_losses	-0.036011	-0.065928	-0.112547	-0.075970	-0.006359	0.623116	1.000

In [21]:

```
sns.heatmap(corr,annot= True)
```

Out[21]:

<Axes: >

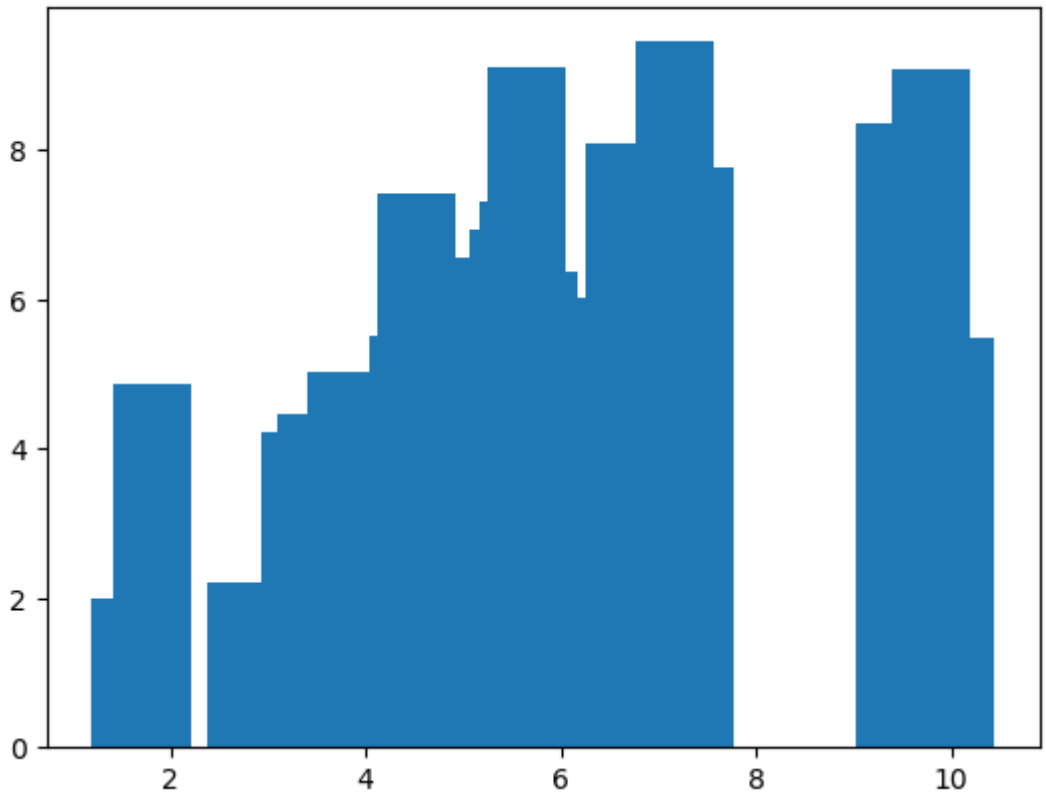


In [22]:

```
x=df["alcohol"]
y = df["speeding"]
plt.bar(x,y)
```

Out[22]:

```
<BarContainer object of 51 artists>
```

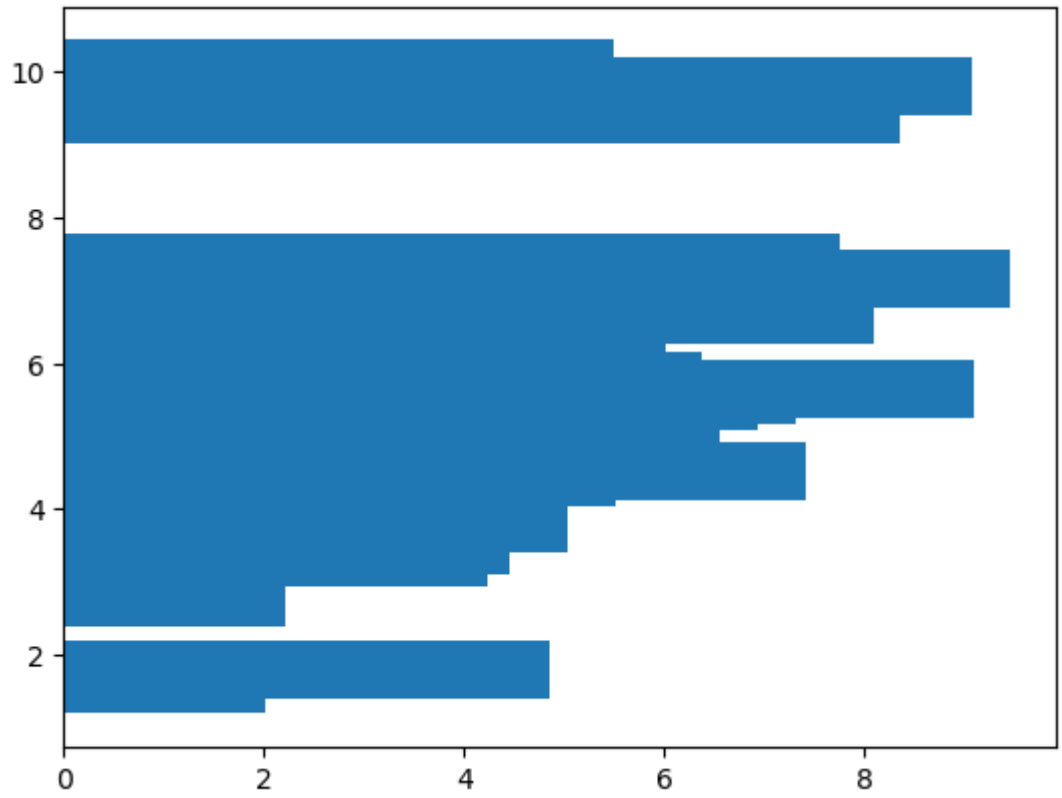


In [23]:

```
plt.barh(x,y)
```

Out[23]:

```
<BarContainer object of 51 artists>
```

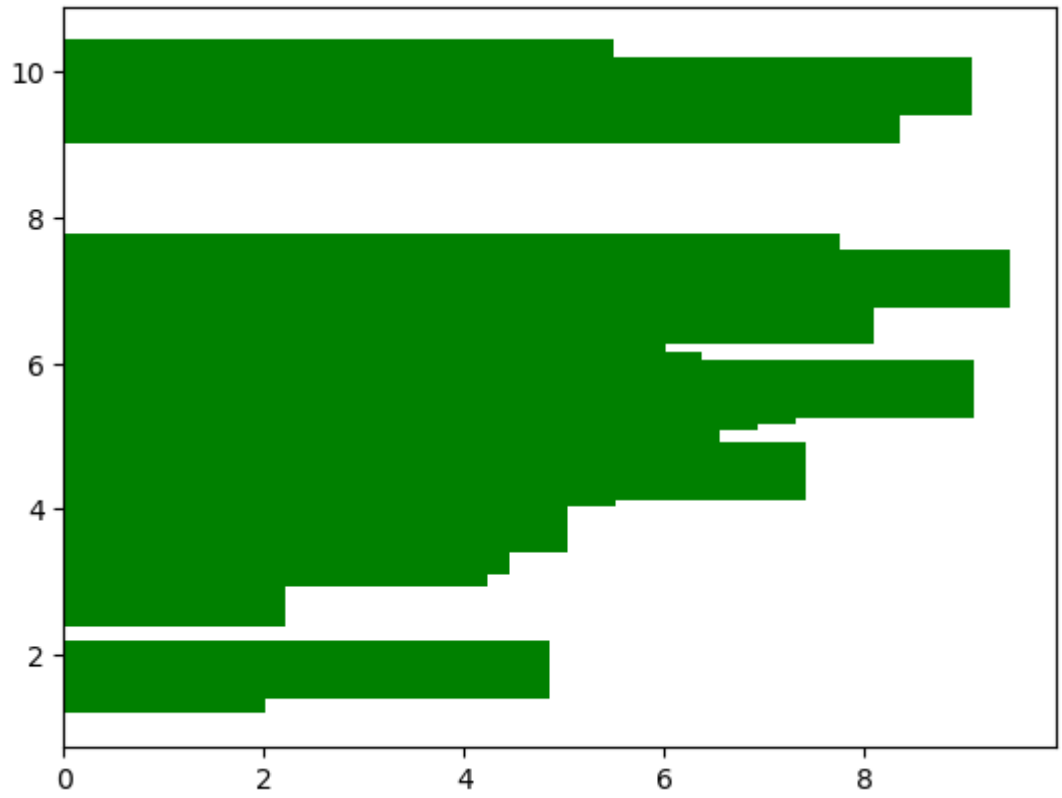


In [24]:

```
plt.barh(x,y,color = 'green')
```

Out[24]:

```
<BarContainer object of 51 artists>
```

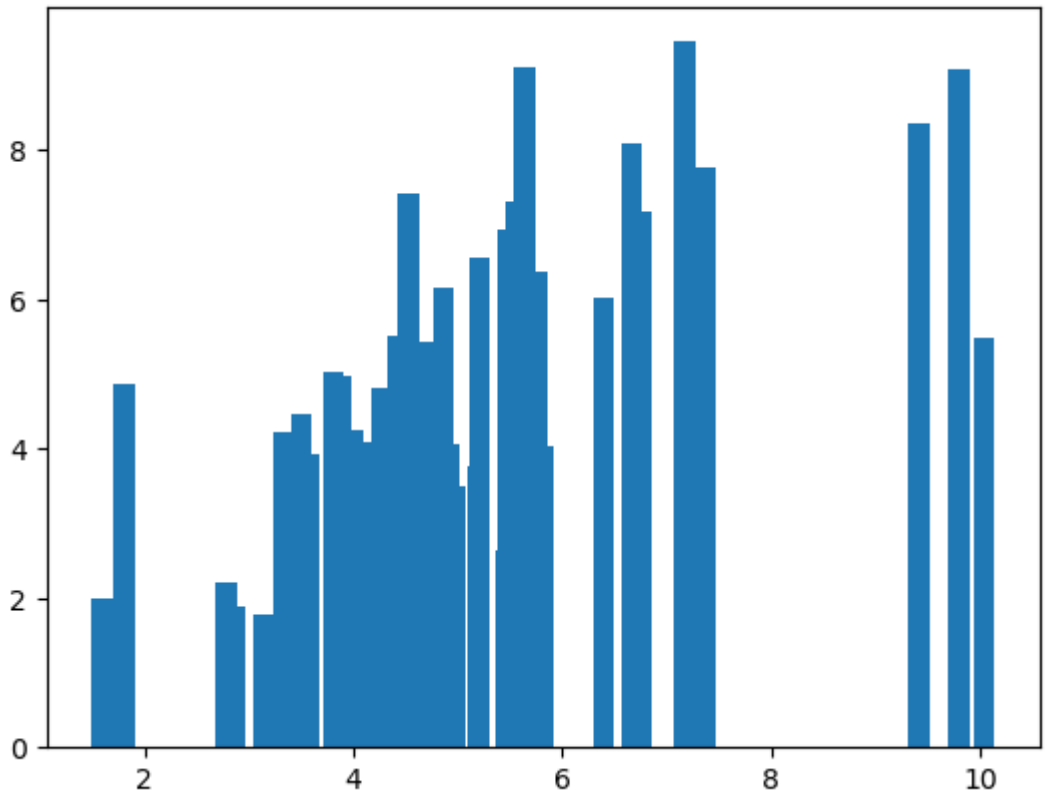


In [25]:

```
plt.bar(x,y,width = 0.2)
```

Out[25]:

```
<BarContainer object of 51 artists>
```

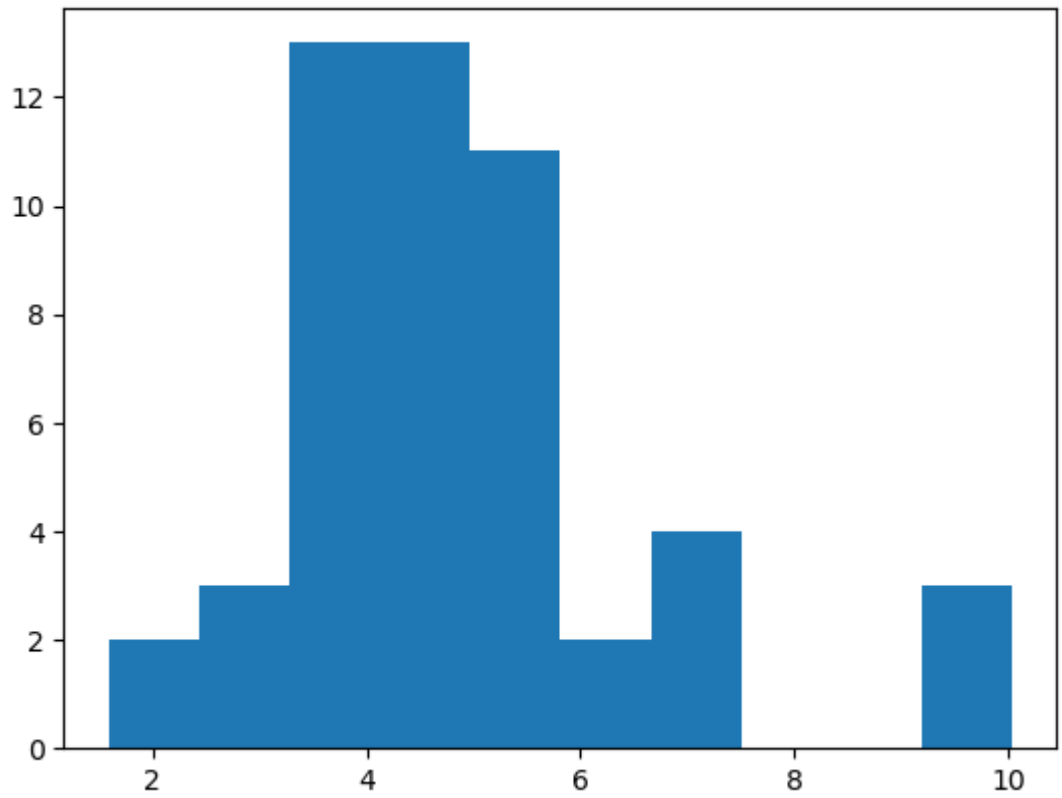


In [26]:

```
plt.hist(x)
```

Out[26]:

```
(array([ 2.,  3., 13., 13., 11.,  2.,  4.,  0.,  0.,  3.]),
 array([ 1.593 ,  2.4375,  3.282 ,  4.1265,  4.971 ,  5.8155,  6.66  ,
         7.5045,  8.349 ,  9.1935, 10.038 ]),
 <BarContainer object of 10 artists>)
```



In [27]:

```
x1 =(df["alcohol"])
fig = plt.figure()
axes1 = fig.add_axes([0.1,0.1,0.8,0.8])
axes1.pie(x1,y,autopct="%0.2f%%",colors=["red","green"])
```

Out[27]:

```
([<matplotlib.patches.Wedge at 0x14b223690>,
  <matplotlib.patches.Wedge at 0x14b278750>,
  <matplotlib.patches.Wedge at 0x14b27a390>,
  <matplotlib.patches.Wedge at 0x14b279a50>,
  <matplotlib.patches.Wedge at 0x14b281c10>,
  <matplotlib.patches.Wedge at 0x14b283910>,
  <matplotlib.patches.Wedge at 0x14b295550>,
  <matplotlib.patches.Wedge at 0x14b297150>,
  <matplotlib.patches.Wedge at 0x14b2a0d90>,
  <matplotlib.patches.Wedge at 0x14b283790>,
  <matplotlib.patches.Wedge at 0x14b2b0550>,
  <matplotlib.patches.Wedge at 0x14b2b2090>,
  <matplotlib.patches.Wedge at 0x14b2b3d10>,
  <matplotlib.patches.Wedge at 0x14b2bda50>,
  <matplotlib.patches.Wedge at 0x14b2bf650>,
  <matplotlib.patches.Wedge at 0x14b2c9310>,
  <matplotlib.patches.Wedge at 0x14b2caf90>,
  <matplotlib.patches.Wedge at 0x14b2d4d90>,
```

In [ ]: