

main

February 4, 2026

```
[38]: import pandas as pd
college = pd.read_csv("College.csv")
college.head(3)
```

```
[38]:
```

	Unnamed: 0	Private	Apps	Accept	Enroll	Top10perc	\
0	Abilene Christian University	Yes	1660	1232	721	23	
1	Adelphi University	Yes	2186	1924	512	16	
2	Adrian College	Yes	1428	1097	336	22	

	Top25perc	F.Undergrad	P.Undergrad	Outstate	Room.Board	Books	Personal	\
0	52	2885	537	7440	3300	450	2200	
1	29	2683	1227	12280	6450	750	1500	
2	50	1036	99	11250	3750	400	1165	

	PhD	Terminal	S.F.Ratio	perc.alumni	Expend	Grad.Rate
0	70	78	18.1	12	7041	60
1	29	30	12.2	16	10527	56
2	53	66	12.9	30	8735	54

```
[39]: college
```

```
[39]:
```

	Unnamed: 0	Private	Apps	Accept	Enroll	Top10perc	\
0	Abilene Christian University	Yes	1660	1232	721	23	
1	Adelphi University	Yes	2186	1924	512	16	
2	Adrian College	Yes	1428	1097	336	22	
3	Agnes Scott College	Yes	417	349	137	60	
4	Alaska Pacific University	Yes	193	146	55	16	
...	
772	Worcester State College	No	2197	1515	543	4	
773	Xavier University	Yes	1959	1805	695	24	
774	Xavier University of Louisiana	Yes	2097	1915	695	34	
775	Yale University	Yes	10705	2453	1317	95	
776	York College of Pennsylvania	Yes	2989	1855	691	28	

	Top25perc	F.Undergrad	P.Undergrad	Outstate	Room.Board	Books	\
0	52	2885	537	7440	3300	450	
1	29	2683	1227	12280	6450	750	
2	50	1036	99	11250	3750	400	

3	89	510	63	12960	5450	450
4	44	249	869	7560	4120	800
..
772	26	3089	2029	6797	3900	500
773	47	2849	1107	11520	4960	600
774	61	2793	166	6900	4200	617
775	99	5217	83	19840	6510	630
776	63	2988	1726	4990	3560	500

	Personal	PhD	Terminal	S.F.Ratio	perc.alumni	Expend	Grad.Rate
0	2200	70	78	18.1	12	7041	60
1	1500	29	30	12.2	16	10527	56
2	1165	53	66	12.9	30	8735	54
3	875	92	97	7.7	37	19016	59
4	1500	76	72	11.9	2	10922	15
..
772	1200	60	60	21.0	14	4469	40
773	1250	73	75	13.3	31	9189	83
774	781	67	75	14.4	20	8323	49
775	2115	96	96	5.8	49	40386	99
776	1250	75	75	18.1	28	4509	99

[777 rows x 19 columns]

```
[40]: college2 = pd.read_csv("College.csv", index_col=0)
college2.head(3)
```

```
[40]:
```

	Private	Apps	Accept	Enroll	Top10perc	\
Abilene Christian University	Yes	1660	1232	721	23	
Adelphi University	Yes	2186	1924	512	16	
Adrian College	Yes	1428	1097	336	22	

	Top25perc	F.Undergrad	P.Undergrad	Outstate	\
Abilene Christian University	52	2885	537	7440	
Adelphi University	29	2683	1227	12280	
Adrian College	50	1036	99	11250	

	Room.Board	Books	Personal	PhD	Terminal	\
Abilene Christian University	3300	450	2200	70	78	
Adelphi University	6450	750	1500	29	30	
Adrian College	3750	400	1165	53	66	

	S.F.Ratio	perc.alumni	Expend	Grad.Rate
Abilene Christian University	18.1	12	7041	60
Adelphi University	12.2	16	10527	56
Adrian College	12.9	30	8735	54

```
[41]: college3 = college.rename({"Unnamed: 0": "College"}, axis=1)
      college3.head(3)
```

```
[41]:
```

	College	Private	Apps	Accept	Enroll	Top10perc	\
0	Abilene Christian University	Yes	1660	1232	721	23	
1	Adelphi University	Yes	2186	1924	512	16	
2	Adrian College	Yes	1428	1097	336	22	

	Top25perc	F.Undergrad	P.Undergrad	Outstate	Room.Board	Books	Personal	\
0	52	2885	537	7440	3300	450	2200	
1	29	2683	1227	12280	6450	750	1500	
2	50	1036	99	11250	3750	400	1165	

	PhD	Terminal	S.F.Ratio	perc.alumni	Expend	Grad.Rate
0	70	78	18.1	12	7041	60
1	29	30	12.2	16	10527	56
2	53	66	12.9	30	8735	54

```
[42]: college3 = college3.set_index("College")
      college3.head(3)
```

```
[42]:
```

	College	Private	Apps	Accept	Enroll	Top10perc	\
	Abilene Christian University	Yes	1660	1232	721	23	
	Adelphi University	Yes	2186	1924	512	16	
	Adrian College	Yes	1428	1097	336	22	

	College	Top25perc	F.Undergrad	P.Undergrad	Outstate	\
	Abilene Christian University	52	2885	537	7440	
	Adelphi University	29	2683	1227	12280	
	Adrian College	50	1036	99	11250	

	College	Room.Board	Books	Personal	PhD	Terminal	\
	Abilene Christian University	3300	450	2200	70	78	
	Adelphi University	6450	750	1500	29	30	
	Adrian College	3750	400	1165	53	66	

	College	S.F.Ratio	perc.alumni	Expend	Grad.Rate
	Abilene Christian University	18.1	12	7041	60
	Adelphi University	12.2	16	10527	56
	Adrian College	12.9	30	8735	54

```
[43]: college = college3
```

```
[44]: college.describe()
```

```
[44]:
```

	Apps	Accept	Enroll	Top10perc	Top25perc	\
count	777.000000	777.000000	777.000000	777.000000	777.000000	
mean	3001.638353	2018.804376	779.972973	27.558559	55.796654	
std	3870.201484	2451.113971	929.176190	17.640364	19.804778	
min	81.000000	72.000000	35.000000	1.000000	9.000000	
25%	776.000000	604.000000	242.000000	15.000000	41.000000	
50%	1558.000000	1110.000000	434.000000	23.000000	54.000000	
75%	3624.000000	2424.000000	902.000000	35.000000	69.000000	
max	48094.000000	26330.000000	6392.000000	96.000000	100.000000	

	F.Undergrad	P.Undergrad	Outstate	Room.Board	Books	\
count	777.000000	777.000000	777.000000	777.000000	777.000000	
mean	3699.907336	855.298584	10440.669241	4357.526384	549.380952	
std	4850.420531	1522.431887	4023.016484	1096.696416	165.105360	
min	139.000000	1.000000	2340.000000	1780.000000	96.000000	
25%	992.000000	95.000000	7320.000000	3597.000000	470.000000	
50%	1707.000000	353.000000	9990.000000	4200.000000	500.000000	
75%	4005.000000	967.000000	12925.000000	5050.000000	600.000000	
max	31643.000000	21836.000000	21700.000000	8124.000000	2340.000000	

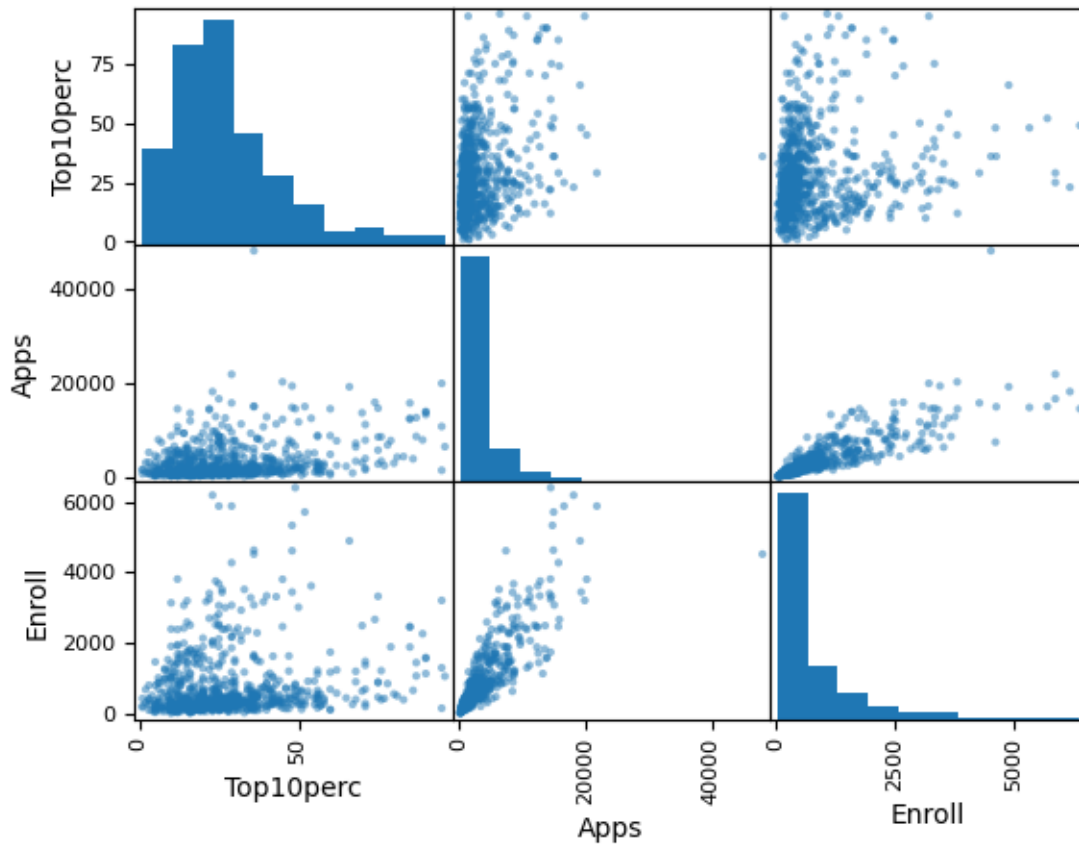
	Personal	PhD	Terminal	S.F.Ratio	perc.alumni	\
count	777.000000	777.000000	777.000000	777.000000	777.000000	
mean	1340.642214	72.660232	79.702703	14.089704	22.743887	
std	677.071454	16.328155	14.722359	3.958349	12.391801	
min	250.000000	8.000000	24.000000	2.500000	0.000000	
25%	850.000000	62.000000	71.000000	11.500000	13.000000	
50%	1200.000000	75.000000	82.000000	13.600000	21.000000	
75%	1700.000000	85.000000	92.000000	16.500000	31.000000	
max	6800.000000	103.000000	100.000000	39.800000	64.000000	

	Expend	Grad.Rate
count	777.000000	777.000000
mean	9660.171171	65.46332
std	5221.768440	17.17771
min	3186.000000	10.00000
25%	6751.000000	53.00000
50%	8377.000000	65.00000
75%	10830.000000	78.00000
max	56233.000000	118.00000

```
[45]: pd.plotting.scatter_matrix(college[["Top10perc", "Apps", "Enroll"]])
```

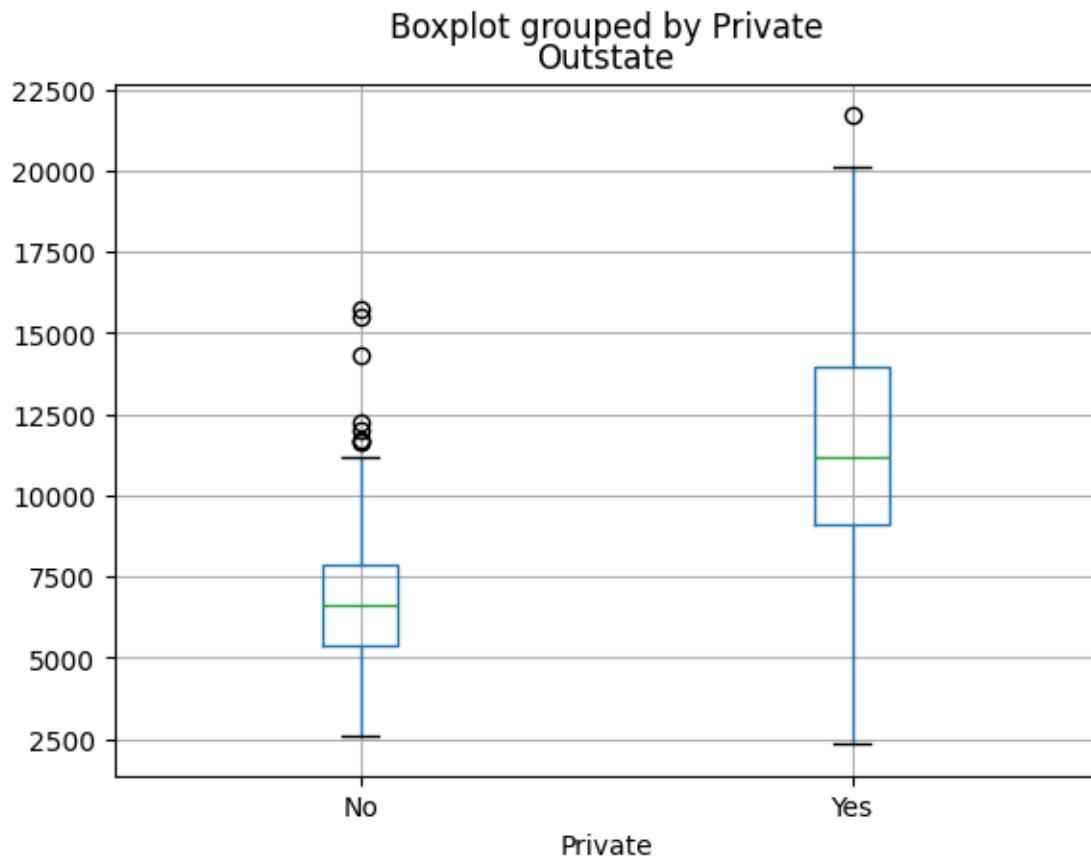
```
[45]: array([[<Axes: xlabel='Top10perc', ylabel='Top10perc'>,
        <Axes: xlabel='Apps', ylabel='Top10perc'>,
        <Axes: xlabel='Enroll', ylabel='Top10perc'>],
```

```
[<Axes: xlabel='Top10perc', ylabel='Apps'>,
 <Axes: xlabel='Apps', ylabel='Apps'>,
 <Axes: xlabel='Enroll', ylabel='Apps'>],
 [<Axes: xlabel='Top10perc', ylabel='Enroll'>,
 <Axes: xlabel='Apps', ylabel='Enroll'>,
 <Axes: xlabel='Enroll', ylabel='Enroll'>]], dtype=object)
```



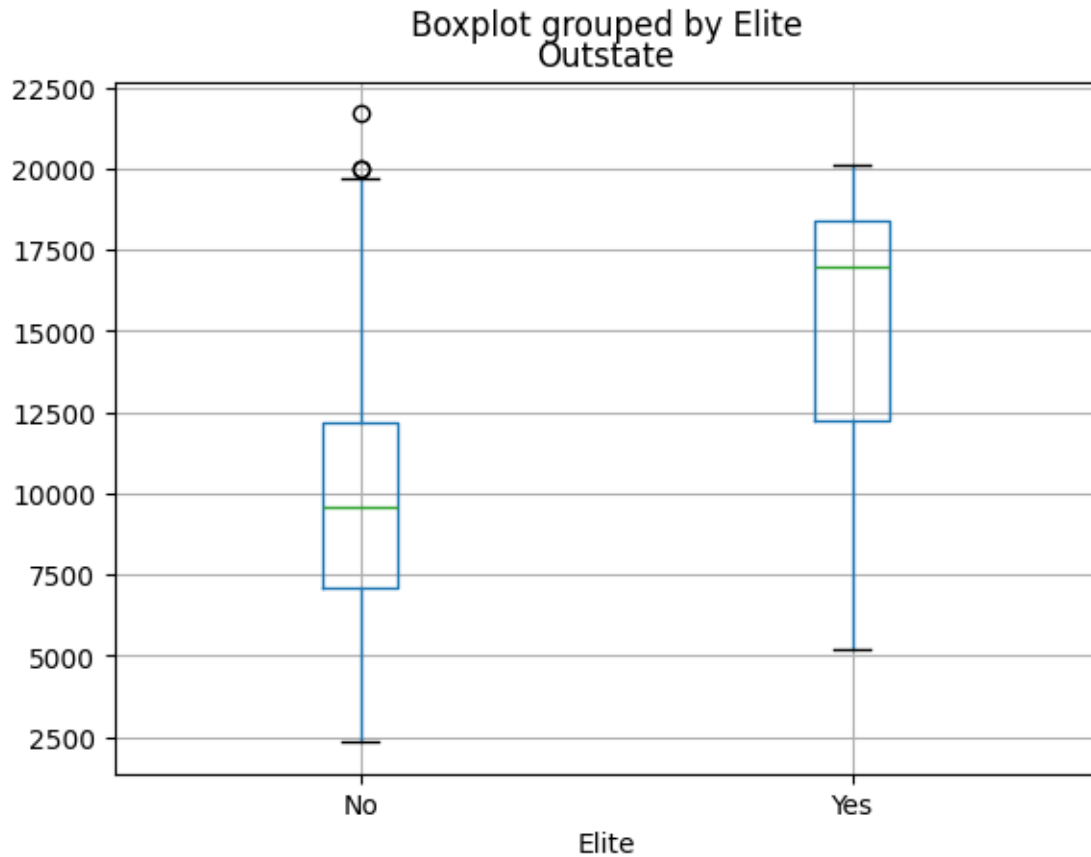
```
[46]: college.boxplot("Outstate", by="Private")
```

```
[46]: <Axes: title={'center': 'Outstate'}, xlabel='Private'>
```



```
[47]: college["Elite"] = pd.cut(college["Top10perc"]/100, [0,0.5,1], labels=["No", "Yes"])
      college["Elite"].value_counts()
      college.boxplot("Outstate", by="Elite")
```

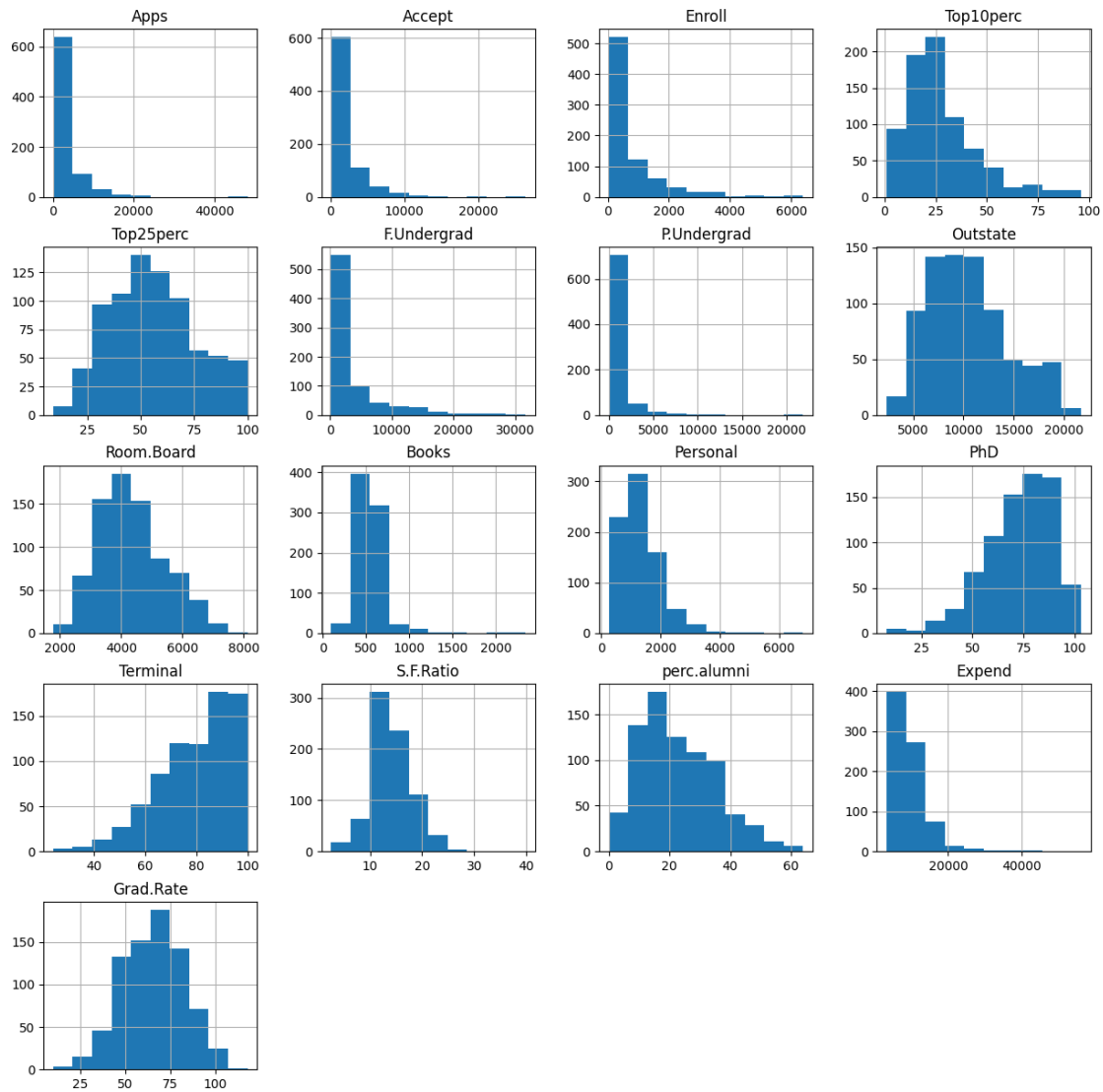
```
[47]: <Axes: title={'center': 'Outstate'}, xlabel='Elite'>
```



```
[48]: college.hist(figsize=(15,15))
```

```
[48]: array([[<Axes: title={'center': 'Apps'}>,
  <Axes: title={'center': 'Accept'}>,
  <Axes: title={'center': 'Enroll'}>,
  <Axes: title={'center': 'Top10perc'}>],
[<Axes: title={'center': 'Top25perc'}>,
  <Axes: title={'center': 'F.Undergrad'}>,
  <Axes: title={'center': 'P.Undergrad'}>,
  <Axes: title={'center': 'Outstate'}>],
[<Axes: title={'center': 'Room.Board'}>,
  <Axes: title={'center': 'Books'}>,
  <Axes: title={'center': 'Personal'}>,
  <Axes: title={'center': 'PhD'}>],
[<Axes: title={'center': 'Terminal'}>,
  <Axes: title={'center': 'S.F.Ratio'}>,
  <Axes: title={'center': 'perc.alumni'}>,
  <Axes: title={'center': 'Expend'}>],
[<Axes: title={'center': 'Grad.Rate'}>, <Axes: >, <Axes: >],
```

<Axes: >]], dtype=object)



```
[49]: auto = pd.read_csv("Auto.csv",na_values=["?"])
auto = auto.dropna()
auto.shape
auto.head(3)
```

```
[49]:   mpg  cylinders  displacement  horsepower  weight  acceleration  year  \
0  18.0         8         307.0         130.0   3504          12.0    70
1  15.0         8         350.0         165.0   3693          11.5    70
2  18.0         8         318.0         150.0   3436          11.0    70
```

origin

name


```

0      1  chevrolet chevelle malibu
1      1      buick skylark 320
2      1      plymouth satellite

```

```
[50]: auto.info()
```

```

<class 'pandas.DataFrame'>
Index: 392 entries, 0 to 396
Data columns (total 9 columns):
#   Column          Non-Null Count  Dtype
---  -
0   mpg             392 non-null   float64
1   cylinders        392 non-null   int64
2   displacement     392 non-null   float64
3   horsepower       392 non-null   float64
4   weight           392 non-null   int64
5   acceleration     392 non-null   float64
6   year            392 non-null   int64
7   origin           392 non-null   int64
8   name            392 non-null   str
dtypes: float64(4), int64(4), str(1)
memory usage: 30.6 KB

```

```
[51]: auto.describe()
```

```

[51]:      mpg  cylinders  displacement  horsepower  weight \
count  392.000000  392.000000    392.000000    392.000000  392.000000
mean    23.445918    5.471939    194.411990    104.469388  2977.584184
std     7.805007    1.705783    104.644004    38.491160   849.402560
min     9.000000    3.000000     68.000000    46.000000  1613.000000
25%    17.000000    4.000000    105.000000    75.000000  2225.250000
50%    22.750000    4.000000    151.000000    93.500000  2803.500000
75%    29.000000    8.000000    275.750000   126.000000  3614.750000
max    46.600000    8.000000   455.000000   230.000000  5140.000000

      acceleration  year  origin
count  392.000000  392.000000  392.000000
mean    15.541327   75.979592   1.576531
std     2.758864    3.683737    0.805518
min     8.000000   70.000000    1.000000
25%    13.775000   73.000000    1.000000
50%    15.500000   76.000000    1.000000
75%    17.025000   79.000000    2.000000
max    24.800000   82.000000    3.000000

```

```
[52]: pd.plotting.scatter_matrix(auto)
```

```

[52]: array([[<Axes: xlabel='mpg', ylabel='mpg'>,
  <Axes: xlabel='cylinders', ylabel='mpg'>,
  <Axes: xlabel='displacement', ylabel='mpg'>,
  <Axes: xlabel='horsepower', ylabel='mpg'>,
  <Axes: xlabel='weight', ylabel='mpg'>,
  <Axes: xlabel='acceleration', ylabel='mpg'>,
  <Axes: xlabel='year', ylabel='mpg'>,
  <Axes: xlabel='origin', ylabel='mpg'>],
[<Axes: xlabel='mpg', ylabel='cylinders'>,
  <Axes: xlabel='cylinders', ylabel='cylinders'>,
  <Axes: xlabel='displacement', ylabel='cylinders'>,
  <Axes: xlabel='horsepower', ylabel='cylinders'>,
  <Axes: xlabel='weight', ylabel='cylinders'>,
  <Axes: xlabel='acceleration', ylabel='cylinders'>,
  <Axes: xlabel='year', ylabel='cylinders'>,
  <Axes: xlabel='origin', ylabel='cylinders'>],
[<Axes: xlabel='mpg', ylabel='displacement'>,
  <Axes: xlabel='cylinders', ylabel='displacement'>,
  <Axes: xlabel='displacement', ylabel='displacement'>,
  <Axes: xlabel='horsepower', ylabel='displacement'>,
  <Axes: xlabel='weight', ylabel='displacement'>,
  <Axes: xlabel='acceleration', ylabel='displacement'>,
  <Axes: xlabel='year', ylabel='displacement'>,
  <Axes: xlabel='origin', ylabel='displacement'>],
[<Axes: xlabel='mpg', ylabel='horsepower'>,
  <Axes: xlabel='cylinders', ylabel='horsepower'>,
  <Axes: xlabel='displacement', ylabel='horsepower'>,
  <Axes: xlabel='horsepower', ylabel='horsepower'>,
  <Axes: xlabel='weight', ylabel='horsepower'>,
  <Axes: xlabel='acceleration', ylabel='horsepower'>,
  <Axes: xlabel='year', ylabel='horsepower'>,
  <Axes: xlabel='origin', ylabel='horsepower'>],
[<Axes: xlabel='mpg', ylabel='weight'>,
  <Axes: xlabel='cylinders', ylabel='weight'>,
  <Axes: xlabel='displacement', ylabel='weight'>,
  <Axes: xlabel='horsepower', ylabel='weight'>,
  <Axes: xlabel='weight', ylabel='weight'>,
  <Axes: xlabel='acceleration', ylabel='weight'>,
  <Axes: xlabel='year', ylabel='weight'>,
  <Axes: xlabel='origin', ylabel='weight'>],
[<Axes: xlabel='mpg', ylabel='acceleration'>,
  <Axes: xlabel='cylinders', ylabel='acceleration'>,
  <Axes: xlabel='displacement', ylabel='acceleration'>,
  <Axes: xlabel='horsepower', ylabel='acceleration'>,
  <Axes: xlabel='weight', ylabel='acceleration'>,
  <Axes: xlabel='acceleration', ylabel='acceleration'>,
  <Axes: xlabel='year', ylabel='acceleration'>,

```

```

    <Axes: xlabel='origin', ylabel='acceleration'>],
[<Axes: xlabel='mpg', ylabel='year'>,
 <Axes: xlabel='cylinders', ylabel='year'>,
 <Axes: xlabel='displacement', ylabel='year'>,
 <Axes: xlabel='horsepower', ylabel='year'>,
 <Axes: xlabel='weight', ylabel='year'>,
 <Axes: xlabel='acceleration', ylabel='year'>,
 <Axes: xlabel='year', ylabel='year'>,
 <Axes: xlabel='origin', ylabel='year'>],
[<Axes: xlabel='mpg', ylabel='origin'>,
 <Axes: xlabel='cylinders', ylabel='origin'>,
 <Axes: xlabel='displacement', ylabel='origin'>,
 <Axes: xlabel='horsepower', ylabel='origin'>,
 <Axes: xlabel='weight', ylabel='origin'>,
 <Axes: xlabel='acceleration', ylabel='origin'>,
 <Axes: xlabel='year', ylabel='origin'>,
 <Axes: xlabel='origin', ylabel='origin'>]], dtype=object)

```

