

```
In [76]: import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
%matplotlib inline
import seaborn as sns
```

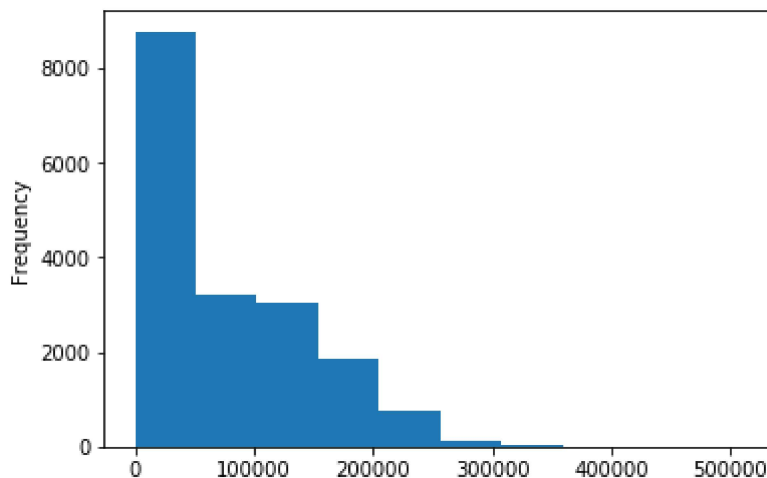
```
In [79]: df=pd.read_csv("./election_results_house.csv")
df
```

District 9	1998	general	1	...	17574.0	28336.0	Greg Harper	DEM	8048.0	16.0
District 9	1998	general	1	...	17575.0	28337.0	Randal Stewart	DEM	2481.0	4.9
District 9	1998	general	1	...	2886.0	28446.0	Barbara Lee	DEM	33497.0	66.8
District 9	1998	general	1	...	13288.0	28447.0	Claiborne Sanders	REP	6114.0	12.1
District 44	1998	general	1	...	17571.0	28333.0	John Overman	REP	1435.0	1.7
District 44	1998	general	1	...	17572.0	28334.0	Tom Harney	REP	1235.0	1.4
District 44	1998	general	1	...	17573.0	28335.0	Amos J.	REP	946.0	1.1

histogram for votes

```
In [66]: v=df["votes"]
v.plot(kind="hist")
```

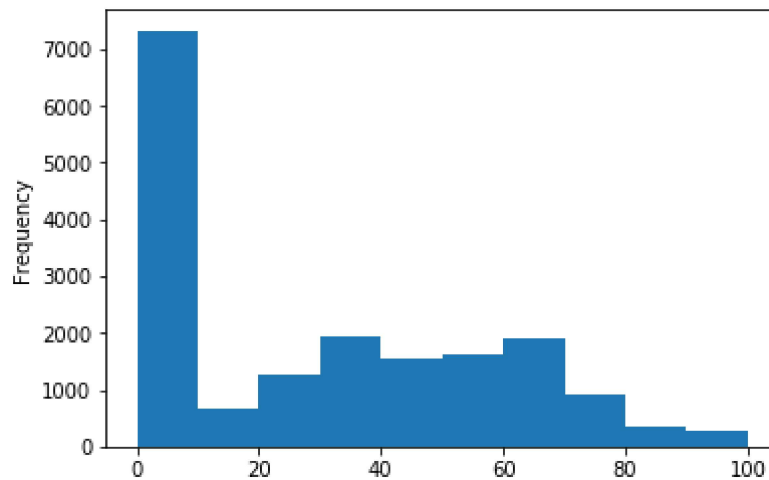
```
Out[66]: <matplotlib.axes._subplots.AxesSubplot at 0x1c3108fc6d8>
```



histogram for percent

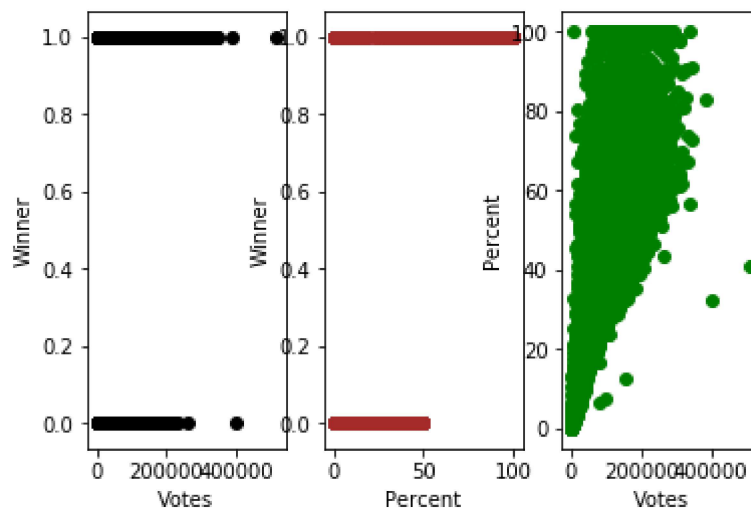
```
In [80]: v=df["percent"]  
v.plot(kind="hist")
```

```
Out[80]: <matplotlib.axes._subplots.AxesSubplot at 0x1c3122b12b0>
```



scatter plot for votes,percent,winner

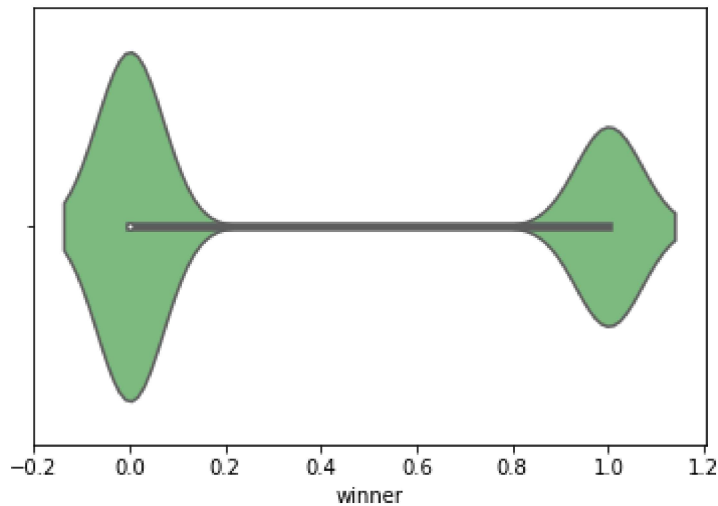
```
In [118]: fig,axes=plt.subplots(nrows=1,ncols=3)  
axes[0].scatter(df.votes,df.winner,color="black")  
axes[0].set_xlabel("Votes")  
axes[0].set_ylabel("Winner")  
axes[1].scatter(df.percent,df.winner,color="brown")  
axes[1].set_xlabel("Percent")  
axes[1].set_ylabel("Winner")  
axes[2].scatter(df.votes,df.percent,color="green")  
axes[2].set_xlabel("Votes")  
axes[2].set_ylabel("Percent")  
plt.show()
```



violin plot for winner

```
In [110]: # from seaborn import set_theme
# sns.set_theme(style="whitegrid")
e=sns.violinplot(x=df["winner"],palette="Greens")
e
```

Out[110]: <matplotlib.axes._subplots.AxesSubplot at 0x1c311ceaeb8>



filling null values

```
In [111]: df1=df.fillna(0)
df1
```

District 1	1998	general	1	...	2735.0	28459.0	Robert A. Brady	DEM	13923.0	73.6	▲
District 1	1998	general	1	...	17569.0	28331.0	Juanita Norwood	REF	1993.0	10.5	
District 1	1998	general	1	...	13902.0	28461.0	William M. Harrison	REP	2436.0	12.8	
District 9	1998	general	1	...	17574.0	28336.0	Greg Harper	DEM	8048.0	16.0	
District 9	1998	general	1	...	17575.0	28337.0	Randal Stewart	DEM	2481.0	4.9	
District 9	1998	general	1	...	2886.0	28446.0	Barbara Lee	DEM	33497.0	66.8	
District 9	1998	general	1	...	13288.0	28447.0	Claiborne Sanders	REP	6114.0	12.7	
District 44	1998	general	1	...	17571.0	28333.0	John Overman	REP	1435.0	1.7	

description of dataset

```
In [114]: df1.describe()
```

Out[114]:

	id	race_id	office_id	cycle	special	party	politician_i
count	17864.000000	17864.000000	17864.000000	17864.000000	17864.000000	17864.0	17864.000000
mean	18679.193798	4054.696037	305.042264	2009.047246	0.041032	0.0	8295.77311
std	7641.566339	2026.535605	165.970047	6.952620	0.198370	0.0	5711.84569
min	3120.000000	167.000000	3.000000	1998.000000	0.000000	0.0	0.000000
25%	14273.750000	2574.750000	163.000000	2002.000000	0.000000	0.0	2978.000000
50%	20416.500000	3980.500000	312.000000	2010.000000	0.000000	0.0	8491.000000
75%	24891.250000	5334.000000	438.000000	2016.000000	0.000000	0.0	13837.000000
max	29621.000000	9478.000000	625.000000	2022.000000	1.000000	0.0	17802.000000

dataset info

```
In [115]: df1.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 17864 entries, 0 to 17863
Data columns (total 21 columns):
id                17864 non-null int64
race_id           17864 non-null int64
state_abbrev      17864 non-null object
state             17864 non-null object
office_id         17864 non-null int64
office_name       17864 non-null object
office_seat_name  17864 non-null object
cycle            17864 non-null int64
stage            17864 non-null object
special           17864 non-null int64
party            17864 non-null float64
politician_id     17864 non-null float64
candidate_id      17864 non-null float64
candidate_name    17864 non-null object
ballot_party      17864 non-null object
votes            17864 non-null float64
percent           17864 non-null float64
unopposed         17864 non-null float64
winner           17864 non-null float64
alt_result_text   17864 non-null object
source            17864 non-null object
dtypes: float64(7), int64(5), object(9)
memory usage: 2.9+ MB
```

defining x&y values and importing logistic regression

```
In [101]: x=df1[['votes','percent']]
          y=df1['winner']
```

```
In [102]: x_train,x_test,y_train,y_test=train_test_split(x,y,test_size=0.60,random_state=0)
```

```
In [103]: from sklearn.model_selection import train_test_split
          from sklearn.linear_model import LogisticRegression
```

```
In [116]: model=LogisticRegression()
          model.fit(x_train,y_train)
```

```
Out[116]: LogisticRegression(C=1.0, class_weight=None, dual=False, fit_intercept=True,
                             intercept_scaling=1, max_iter=100, multi_class='ovr', n_jobs=1,
                             penalty='l2', random_state=None, solver='liblinear', tol=0.0001,
                             verbose=0, warm_start=False)
```

predictions and classification

```
In [105]: predictions=model.predict(x_test)
          predictions
```

```
Out[105]: array([1., 1., 0., ..., 1., 1., 1.])
```

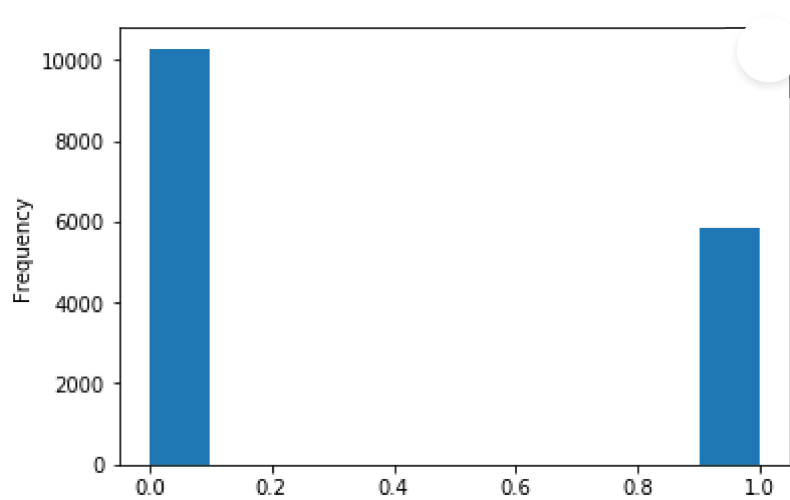
```
In [121]: from sklearn.metrics import classification_report
          print(classification_report(y_test,predictions))
```

	precision	recall	f1-score	support
0.0	0.96	0.39	0.56	7182
1.0	0.44	0.96	0.60	3537
avg / total	0.79	0.58	0.57	10719

histogram for winner

```
In [120]: v=df["winner"]  
v.plot(kind="hist")
```

```
Out[120]: <matplotlib.axes._subplots.AxesSubplot at 0x1c30f7abef0>
```



accuracy

```
In [108]: from sklearn.metrics import accuracy_score  
accuracy_score(y_test,predictions)
```

```
Out[108]: 0.5805578878626737
```

```
In [ ]:
```