Pulse Sales

December 10, 2017

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

```
/usr/local/lib/python2.7/dist-packages/pandas/core/computation/__init__.py:18: UserWarning: The
The minimum supported version is 2.4.6

  ver=ver, min_ver=_MIN_NUMEXPR_VERSION), UserWarning)
```

```
In [73]: pulse = pd.read_csv('Pulse.csv')[1:]
```

```
In [75]: pulse.head()
```

Out[75]:

	Sales	Advertising	deltaa	profit
1	71.486107	1303.990603	143.887845	-1.303991e+05
2	24.178528	0.100000	-0.999923	-1.000000e+01
3	30038.896999	10000.000021	99999.000209	-1.000000e+06
4	12011.142884	0.100000	-0.999990	-1.000000e+01
5	34833.682742	10000.000021	99999.000209	-1.000000e+06

```
In [77]: pulse.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 35 entries, 1 to 35
Data columns (total 4 columns):
Sales          35 non-null float64
Advertising    35 non-null float64
deltaa         35 non-null float64
profit         35 non-null float64
dtypes: float64(4)
memory usage: 1.2 KB
```

```
In [78]: pulse.describe()
```

Out[78]:

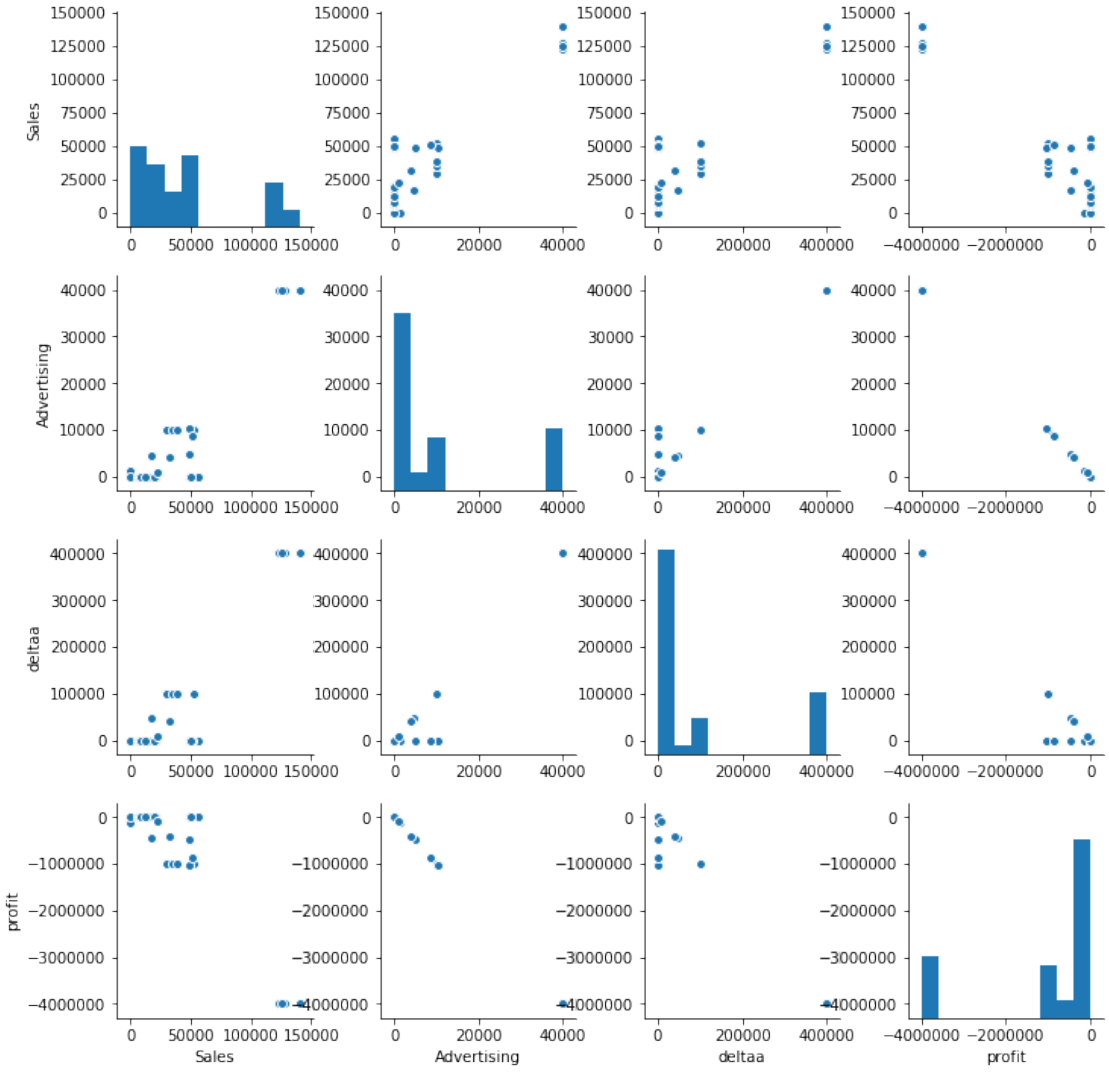
	Sales	Advertising	deltaa	profit
count	35.000000	35.000000	35.000000	3.500000e+01
mean	46664.924151	10136.800509	94145.501278	-1.013680e+06
std	43798.387142	15577.713110	158446.183020	1.557771e+06
min	24.178528	0.100000	-0.999997	-4.000000e+06
25%	14653.875206	0.100000	-0.999977	-1.000000e+06
50%	31731.144526	892.127339	-0.879447	-8.921273e+04
75%	51813.707392	10000.000021	99999.000208	-1.000000e+01
max	140128.335773	40000.000000	399999.000000	-1.000000e+01

In [79]: pulse.columns

Out[79]: Index([u'Sales', u'Advertising', u'deltaa', u'profit'], dtype='object')

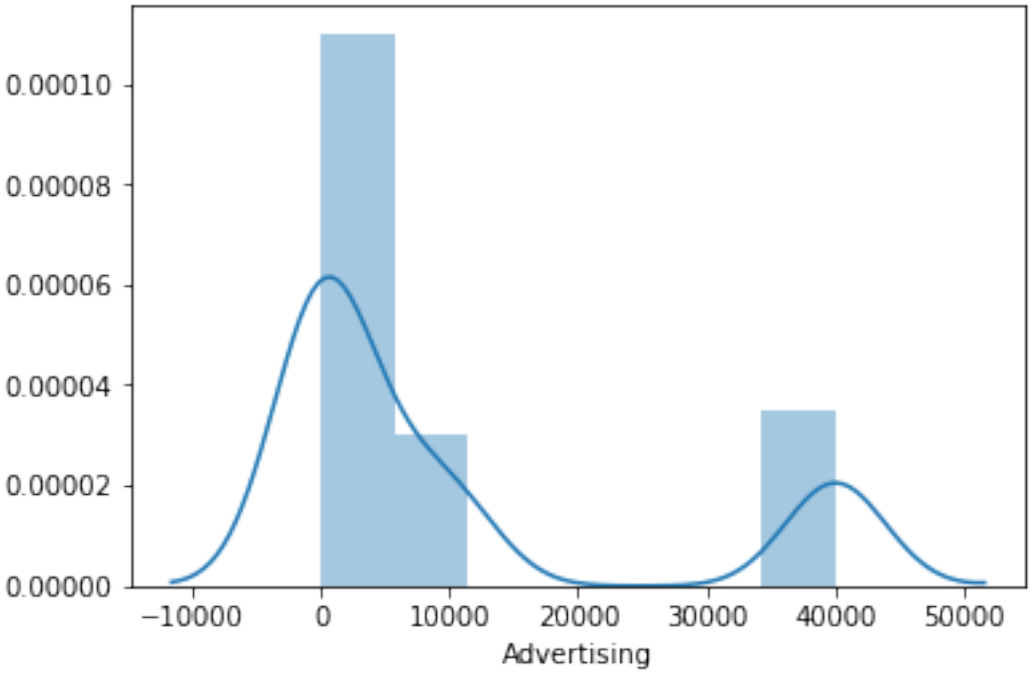
In [80]: sns.pairplot(pulse)

Out[80]: <seaborn.axisgrid.PairGrid at 0x7fbd17d12150>



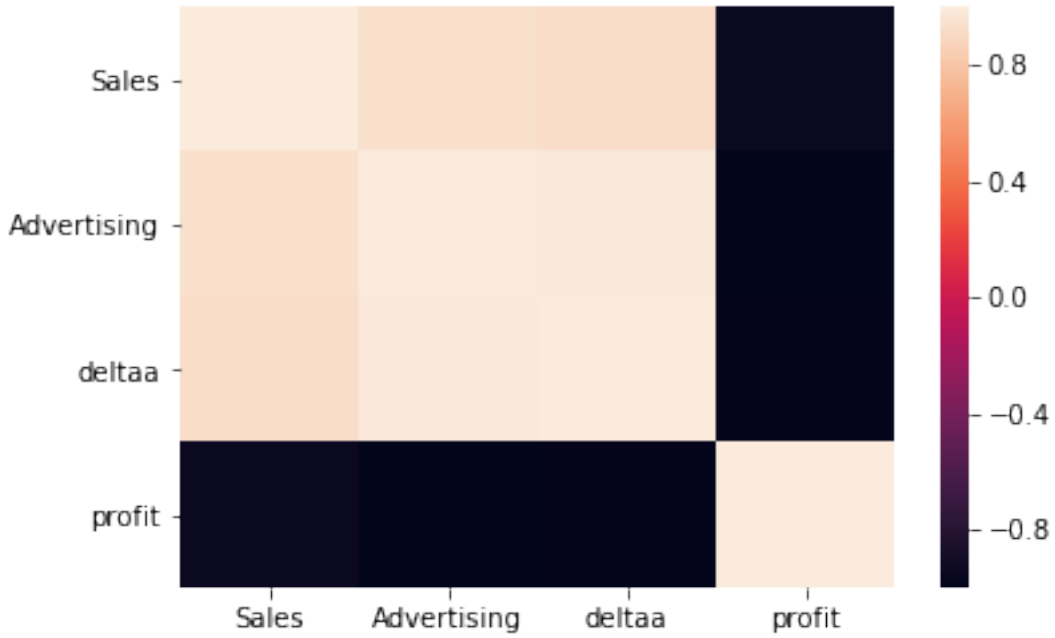
In [81]: sns.distplot(pulse['Advertising'])

Out[81]: <matplotlib.axes._subplots.AxesSubplot at 0x7fbd174e8e90>



In [82]: sns.heatmap(pulse.corr())

Out[82]: <matplotlib.axes._subplots.AxesSubplot at 0x7fbd1751a1d0>

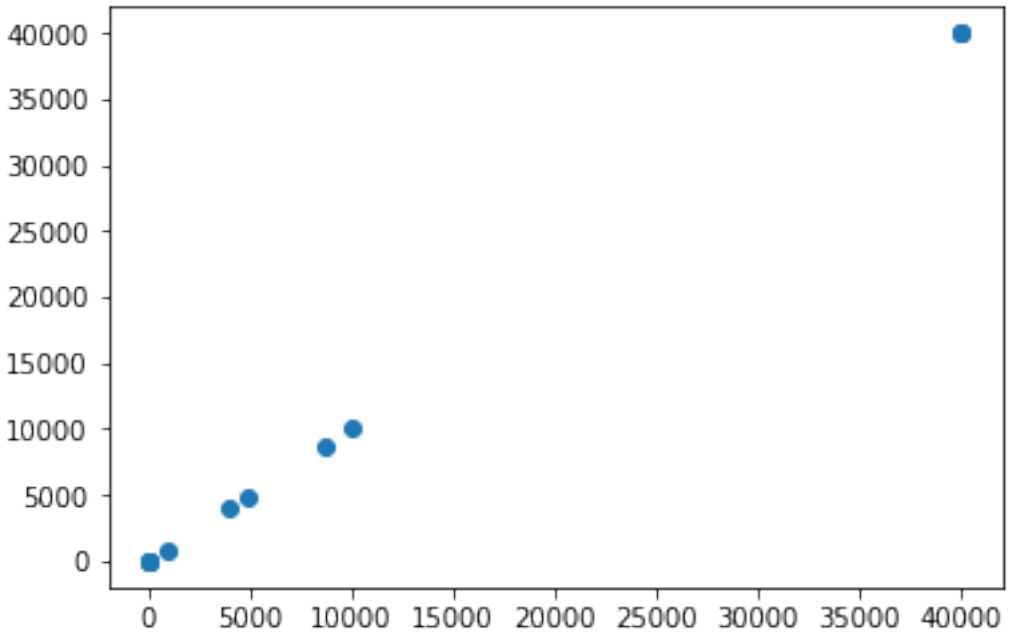



```
In [83]: X = pulse.drop('Advertising', axis=1)
         y = pulse['Advertising']

In [84]: from sklearn.model_selection import train_test_split

In [85]: X_train, X_test, y_train, y_test = train_test_split(X, y, test_size= 0.33, random_state

In [86]: from sklearn.linear_model import LinearRegression

In [87]: lm = LinearRegression()

In [88]: lm.fit(X_train,y_train)

Out[88]: LinearRegression(copy_X=True, fit_intercept=True, n_jobs=1, normalize=False)

In [89]: print(lm.intercept_)

1.81898940355e-12


In [90]: coeff_df = pd.DataFrame(lm.coef_,X.columns,columns=['Coefficient'])
         coeff_df

Out[90]:          Coefficient
         Sales  -9.810027e-17
         deltaa  6.938894e-17
         profit -1.000000e-02
```

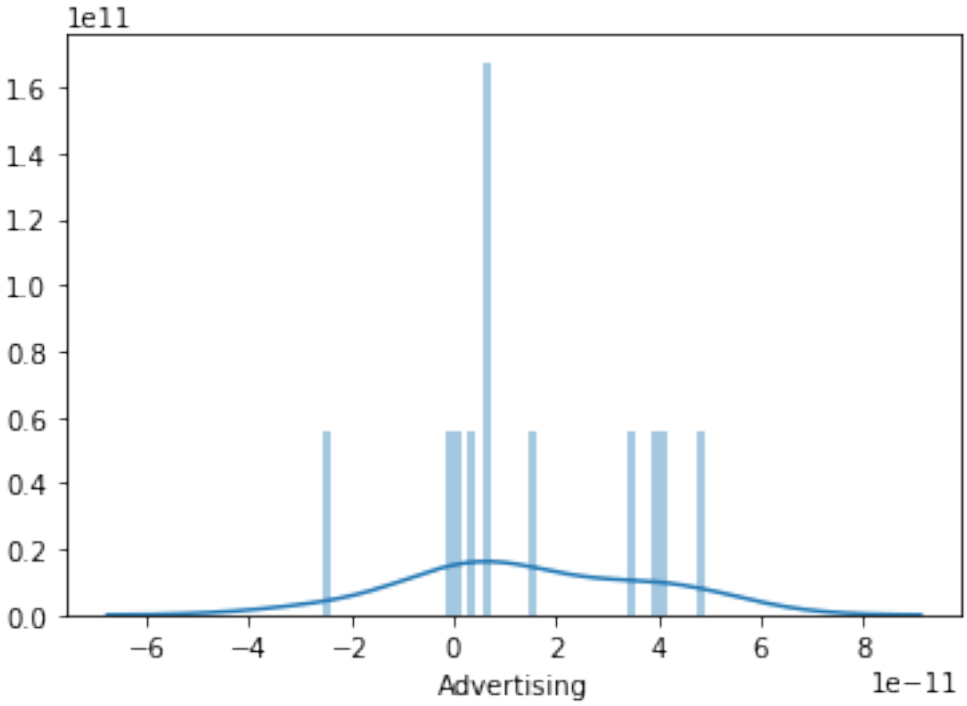
```
In [91]: predictions = lm.predict(X_test)
```

```
In [92]: plt.scatter(y_test,predictions)
```

```
Out[92]: <matplotlib.collections.PathCollection at 0x7fbd17384a90>
```




```
In [93]: sns.distplot((y_test-predictions),bins=50);
```



```
In [94]: from sklearn import metrics

In [95]: print('MAE:', metrics.mean_absolute_error(y_test, predictions))
         print('MSE:', metrics.mean_squared_error(y_test, predictions))
         print('RMSE:', np.sqrt(metrics.mean_squared_error(y_test, predictions)))

('MAE:', 1.9205411566636243e-11)
('MSE:', 6.5544330976620767e-22)
('RMSE:', 2.5601627092163649e-11)
```