ProjectLR

July 19, 2017

In [2]:
```
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
%matplotlib inline
import seaborn as sns
```

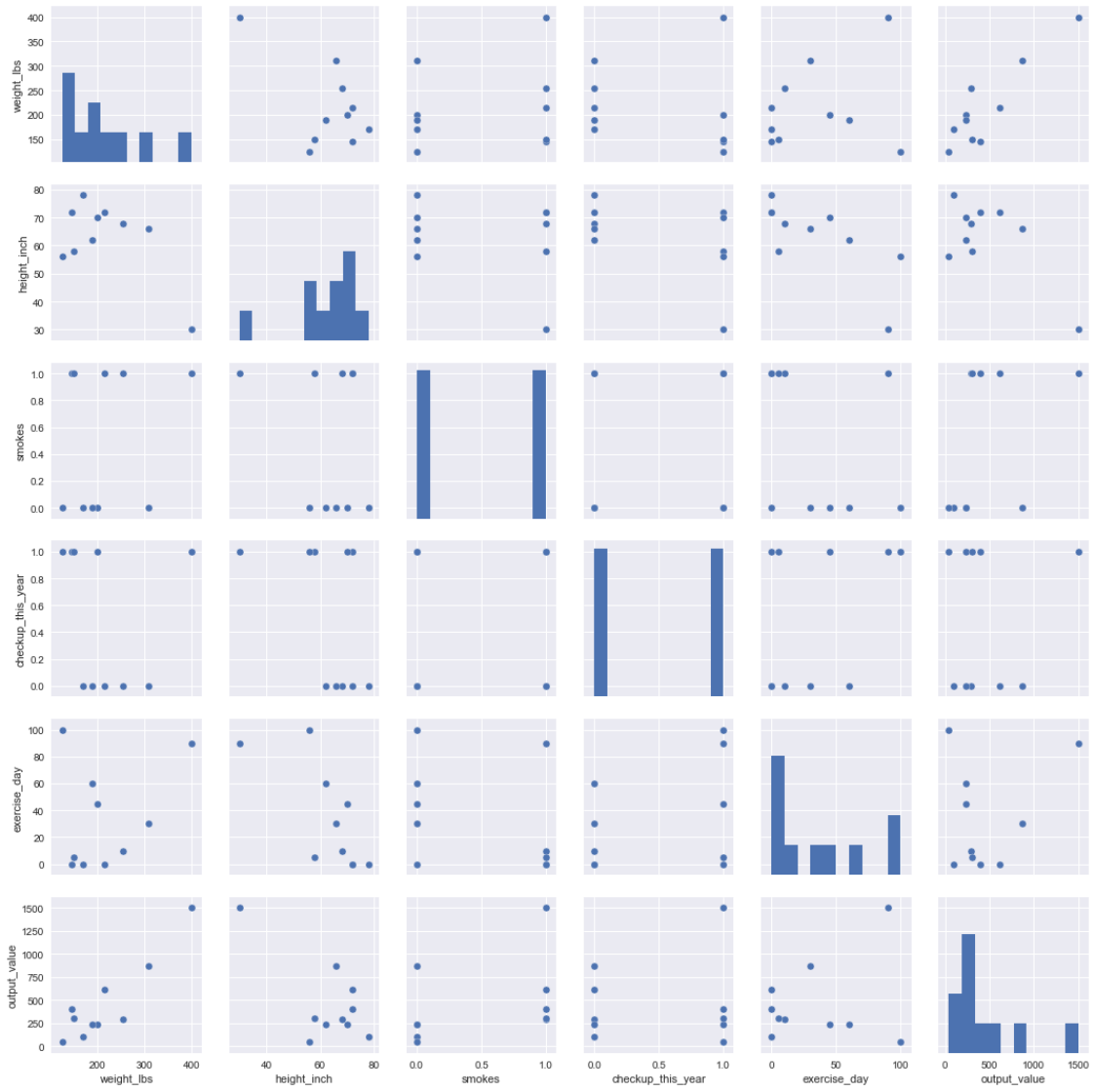
In [15]:
```
# Process data
df = pd.read_csv('patient.csv')
df.drop('person_id',axis=1,inplace=True)

df
```

Out[15]:

	weight_lbs	height_inch	smokes	checkup_this_year	exercise_day	output_value
0	145	72	1	1	0	400
1	200	70	0	1	45	240
2	255	68	1	0	10	290
3	310	66	0	0	30	870
4	170	78	0	0	0	100
5	190	62	0	0	60	240
6	215	72	1	0	0	620
7	150	58	1	1	5	300
8	125	56	0	1	100	45
9	400	30	1	1	90	1500

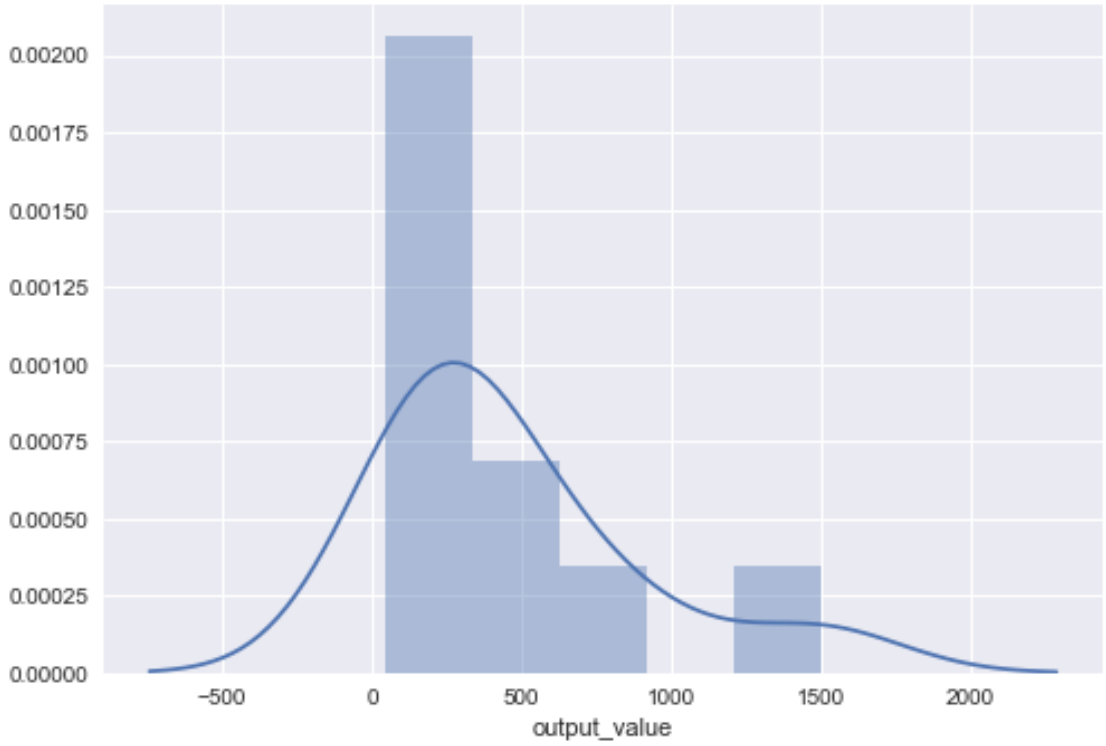
In [16]:
```
sns.pairplot(df)
```

Out[16]: <seaborn.axisgrid.PairGrid at 0x7dcc564438>



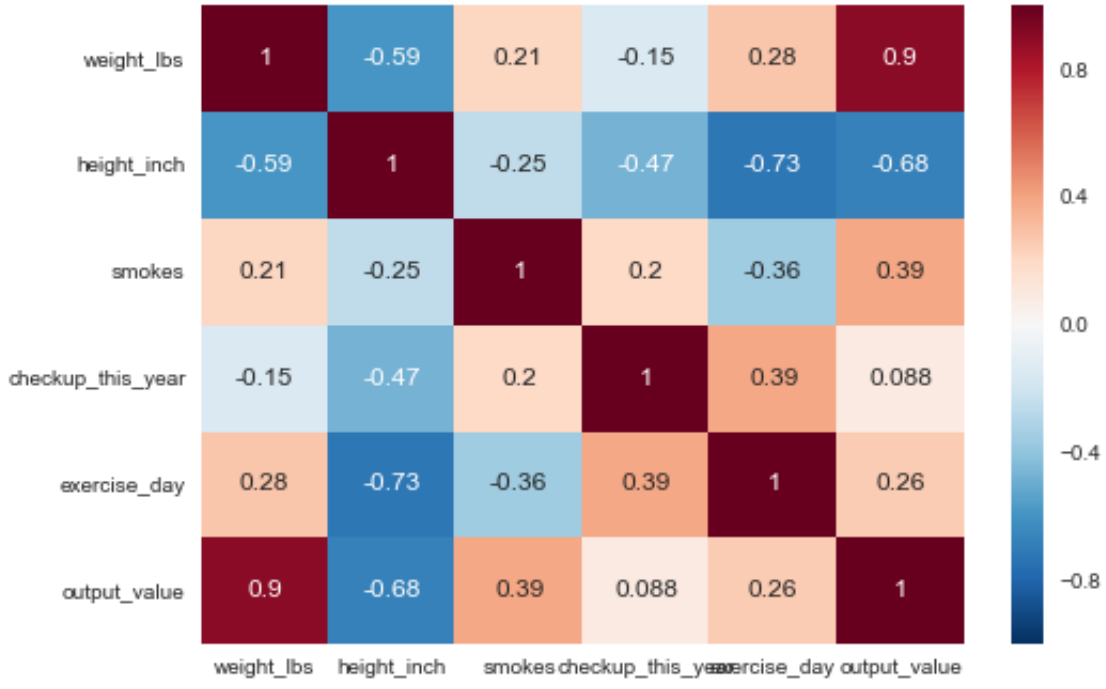
In [17]: sns.distplot(df['output_value'])

Out[17]: <matplotlib.axes._subplots.AxesSubplot at 0x7dceb938d0>

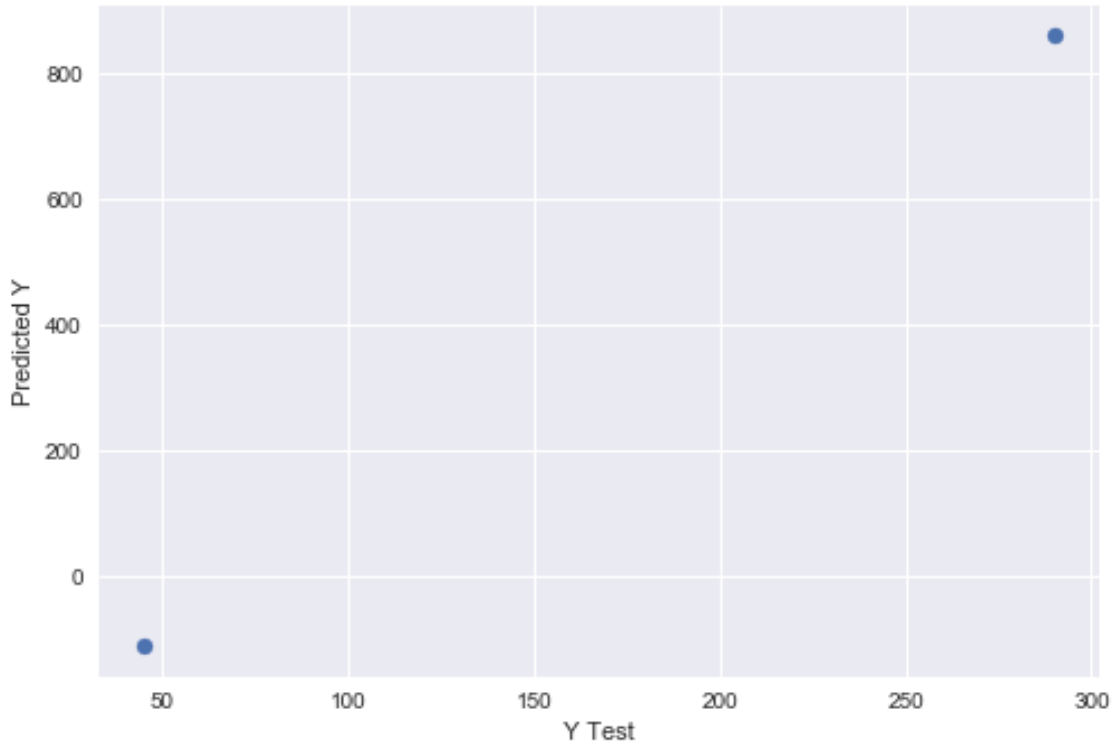



```
In [18]: sns.heatmap(df.corr(), annot=True)
```

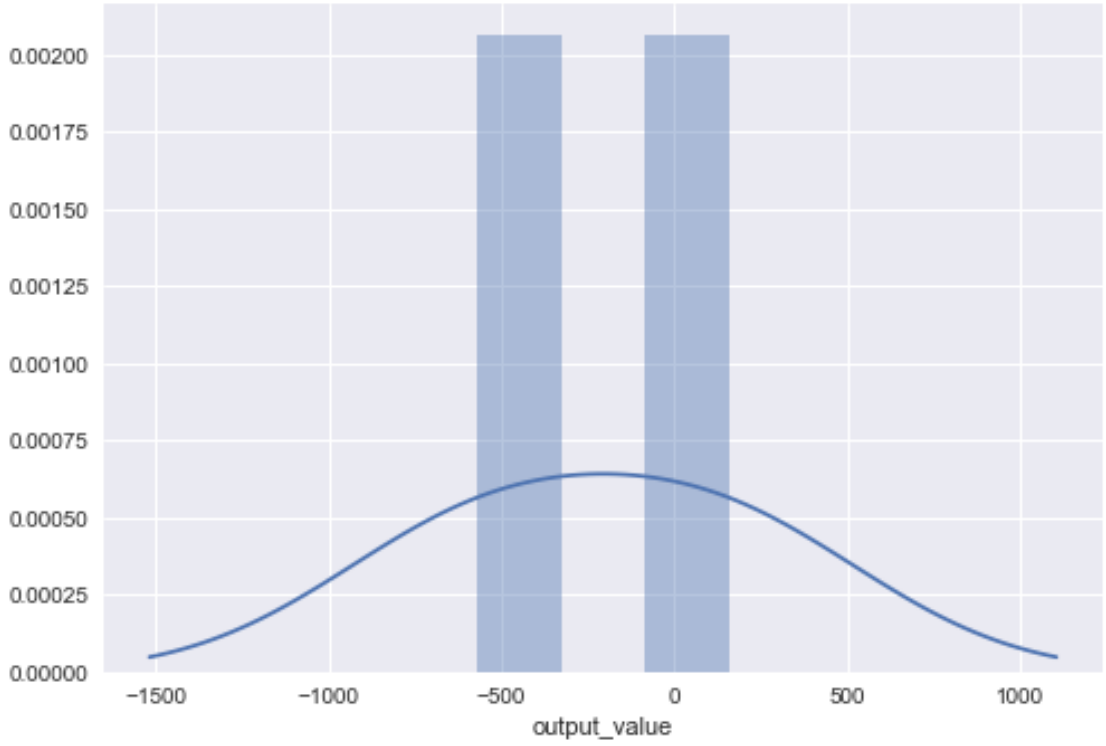
```
Out[18]: <matplotlib.axes._subplots.AxesSubplot at 0x7dcecfe630>
```



```
In [19]: # Set input and output
         X = df[['weight_lbs', 'height_inch', 'smokes', 'checkup_this_year', 'exercise_day']]
         y = df['output_value']

In [21]: from sklearn.model_selection import train_test_split

In [22]: X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.2, random_state

In [23]: from sklearn.linear_model import LinearRegression

         lm = LinearRegression()

         lm.fit(X_train,y_train)

Out[23]: LinearRegression(copy_X=True, fit_intercept=True, n_jobs=1, normalize=False)

In [24]: predictions = lm.predict(X_test)

In [25]: plt.scatter(y_test,predictions)
         plt.xlabel('Y Test')
         plt.ylabel('Predicted Y')

Out[25]: <matplotlib.text.Text at 0x7dcfe21668>
```



```
In [40]: sns.distplot((y_test-predictions),bins=3)

Out[40]: <matplotlib.axes._subplots.AxesSubplot at 0x7dcfeeefd0>
```




```
In [27]: from sklearn import metrics

In [28]: 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: 363.472311339
MSE: 174788.715653
RMSE: 418.077403901


In [29]: coeff_diff = pd.DataFrame(lm.coef_,X.columns,columns=['Coefficient'])
         coeff_diff

Out[29]:                    Coefficient
         weight_lbs            5.017399
         height_inch           2.795952
```

```
smokes              321.370216
checkup_this_year     9.207331
exercise_day          0.289239
```

Interpreting the coefficients:

Holding all other features fixed, a 1 unit increase in weight_lbs is associated with an increase of 5.017399 . Holding all other features fixed, a 1 unit increase in height_inch is associated with an increase of 2.795952 . Holding all other features fixed, a 1 unit increase in smokes is associated with an increase of 321.370216 . Holding all other features fixed, a 1 unit increase in checkup_this_year is associated with an increase of 9.207331 . Holding all other features fixed, a 1 unit increase in exercide_day is associated with an increase of 0.289239 .