In [2]:

```
import numpy as np
import pandas as pd
from sklearn import datasets
from sklearn.datasets import load_boston
from sklearn import linear_model
from sklearn import preprocessing
from sklearn.preprocessing import PolynomialFeatures
from sklearn.model_selection import train_test_split
from sklearn.metrics import mean_squared_error, mean_absolute_error, r2_score
import matplotlib.pyplot as plt
import matplotlib.ticker as ticker
import seaborn as sns
```

In [4]:

```
np.random.seed(42)
```

Importing Data Set

In [28]:

```
cement_df = pd.read_csv('homework3_input_data.csv')
cement_df
```

Out[28]:

	cement	slag	flyash	water	superplasticizer	coarseaggregate	fineaggregate	age	csMP
0	540.0	0.0	0.0	162.0	2.5	1040.0	676.0	28	79.9
1	540.0	0.0	0.0	162.0	2.5	1055.0	676.0	28	61.8
2	332.5	142.5	0.0	228.0	0.0	932.0	594.0	270	40.2
3	332.5	142.5	0.0	228.0	0.0	932.0	594.0	365	41.0
4	198.6	132.4	0.0	192.0	0.0	978.4	825.5	360	44.3
...	...	...	...	...	...	...	...	...	.
1025	276.4	116.0	90.3	179.6	8.9	870.1	768.3	28	44.2
1026	322.2	0.0	115.6	196.0	10.4	817.9	813.4	28	31.1
1027	148.5	139.4	108.6	192.7	6.1	892.4	780.0	28	23.7
1028	159.1	186.7	0.0	175.6	11.3	989.6	788.9	28	32.7
1029	260.9	100.5	78.3	200.6	8.6	864.5	761.5	28	32.4

1030 rows × 9 columns

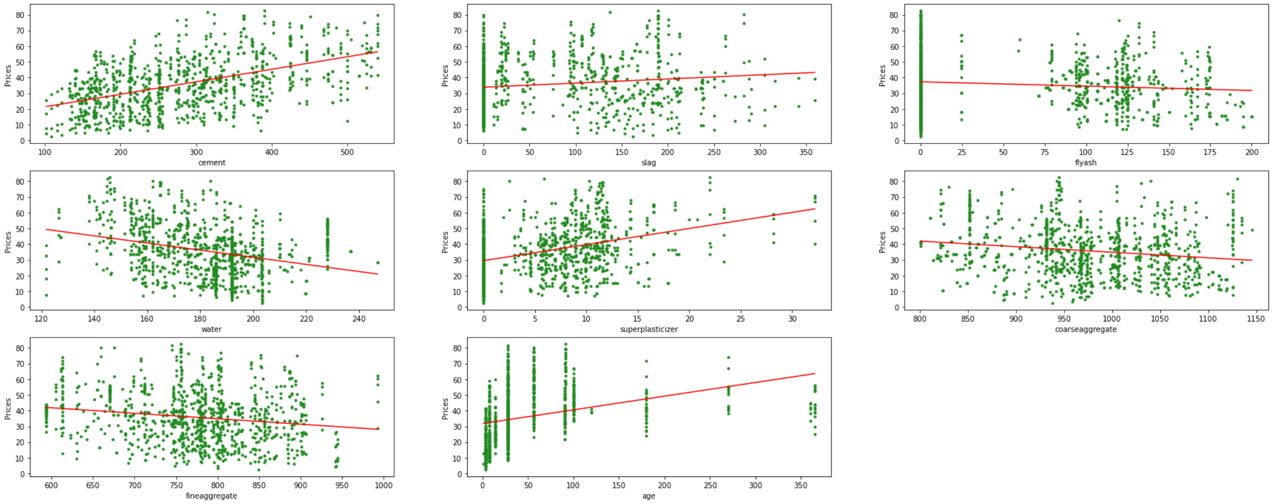
Splitting Dataset

In [17]:

```
X_train, X_test, Y_train, Y_test = train_test_split(cement_df[['cement','slag','
```

Linear Regression Line for Data

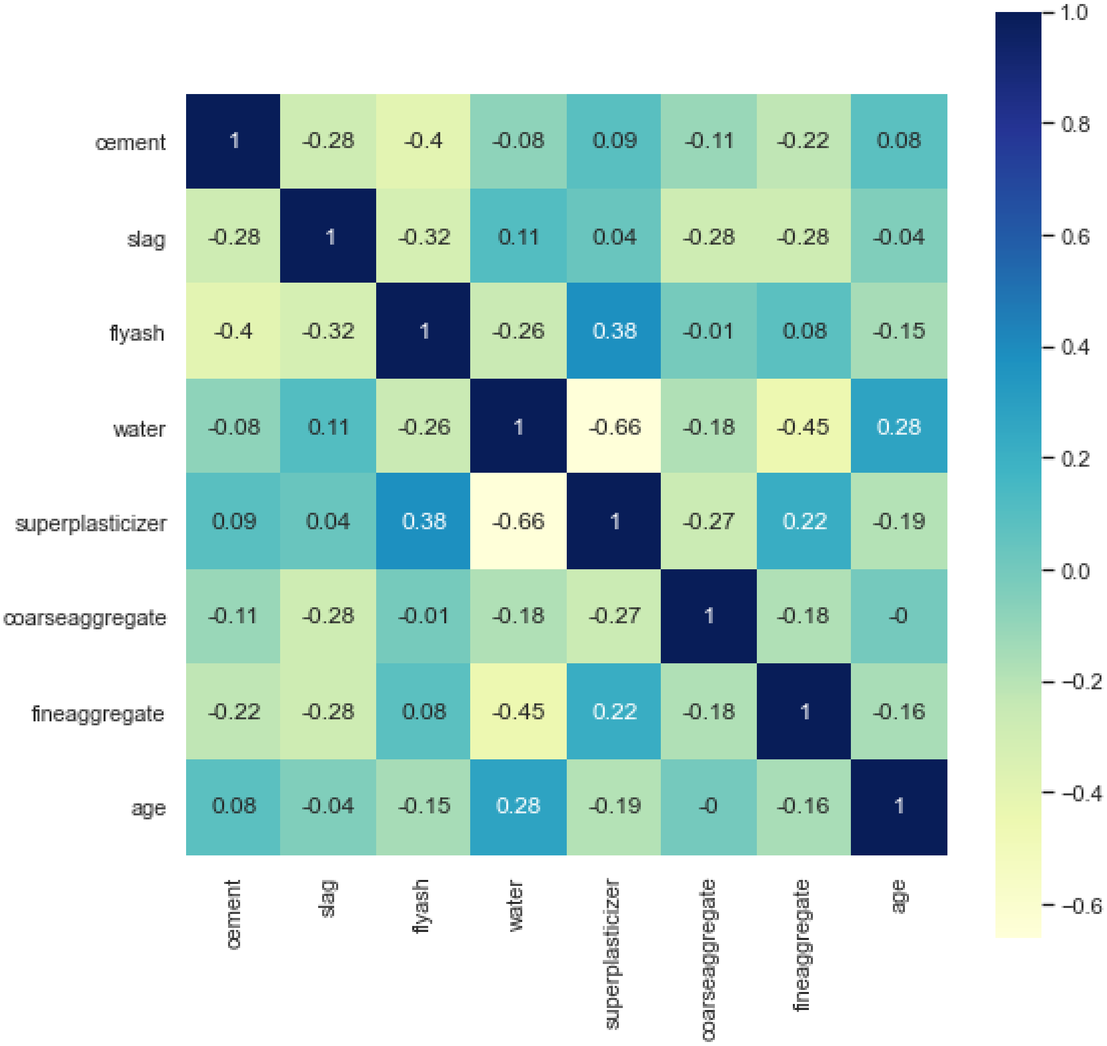
In [18]:
```
plt.figure(figsize=(30,20))
for i, col in enumerate(cement_df.columns[0:8]):
    plt.subplot(5, 3, i+1)
    x = cement_df[col]
    y = cement_df['csMPa']
    plt.plot(x, y, '.', color="forestgreen")
    # create linear regression line:
    plt.plot(np.unique(x), np.poly1d(np.polyfit(x, y, 1))(np.unique(x)),color="r
    plt.xlabel(col)
    plt.ylabel('Cement')
```



Confusion Matrix

In [19]:
```
features = cement_df[['cement','slag','flyash','water','superplasticizer','coars
sns.set(rc={'figure.figsize': (8.5,8.5)})
sns.heatmap(features.corr().round(2), square=True, cmap='YlGnBu', annot=True)
```

Out[19]:
```
<AxesSubplot:>
```



In [20]:

```
X_train.shape, Y_train.shape, X_test.shape, Y_test.shape
```

Out[20]:

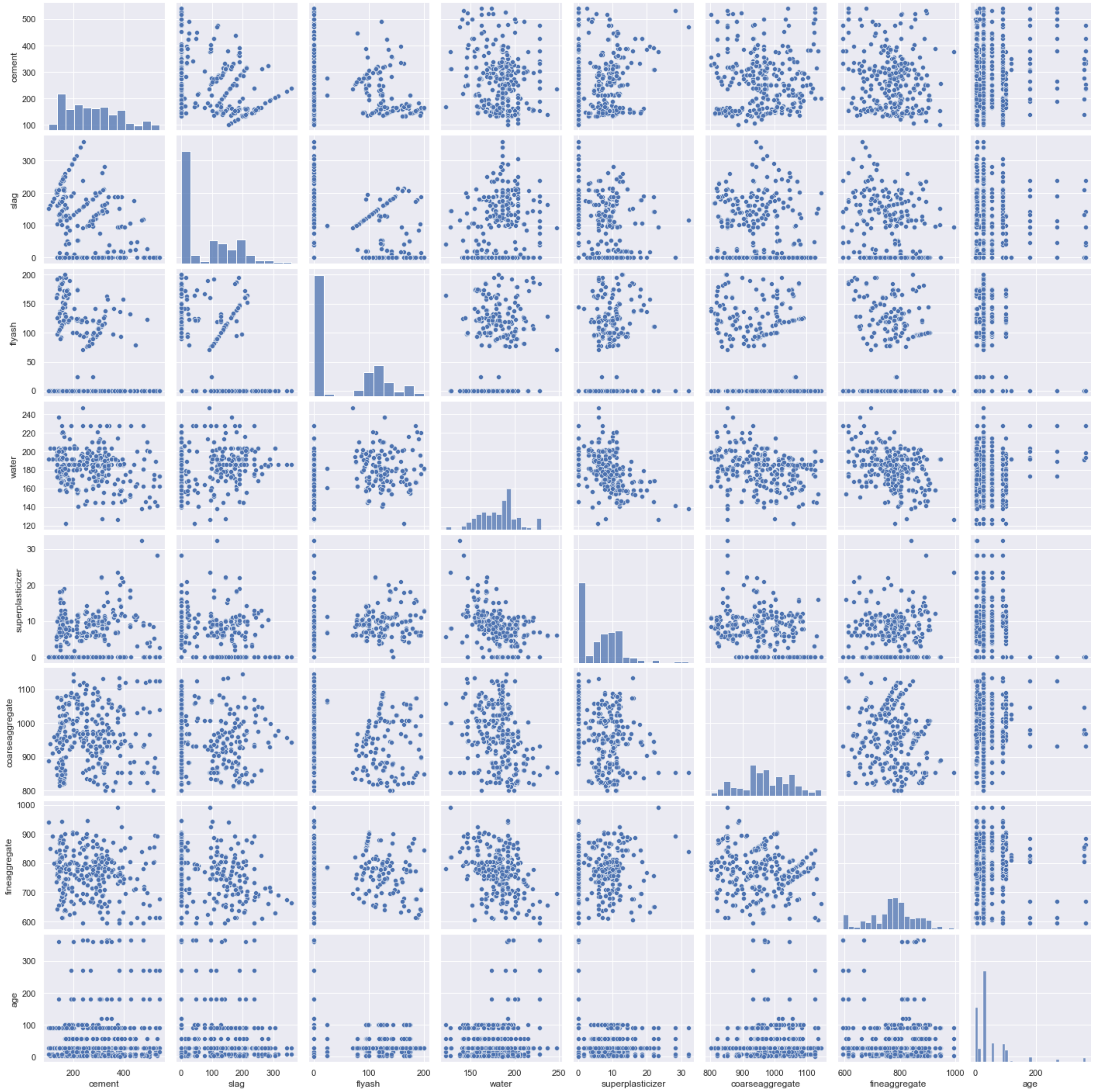
```
((824, 8), (824,), (206, 8), (206,))
```

In [24]:

```
train_df = pd.DataFrame(X_train,columns = ['cement','slag','flyash','water','sup
train_df['csMPa'] = Y_train
sns.pairplot(train_df, vars = ['cement','slag','flyash','water','superplasticize
```

Out[24]:

```
<seaborn.axisgrid.PairGrid at 0x7f9fd0d3f670>
```



Linear Regression Model

In [25]:

```
model = linear_model.LinearRegression().fit(X_train, Y_train)
```

In [26]:

```
# The coefficients:
print('Coefficients: \n', model.coef_)

Y_test_pred = model.predict(X_test)

# The mean squared error:
print('Mean squared error: %.2f' % mean_squared_error(Y_test, Y_test_pred))

# The coefficient of determination (1 is perfect prediction):
print('Coefficient of determination: %.2f' % r2_score(Y_test, Y_test_pred))
```

```
Coefficients: 
 [ 0.11923772  0.10881555  0.0911555  -0.14527714  0.31551104  0.02225423
```

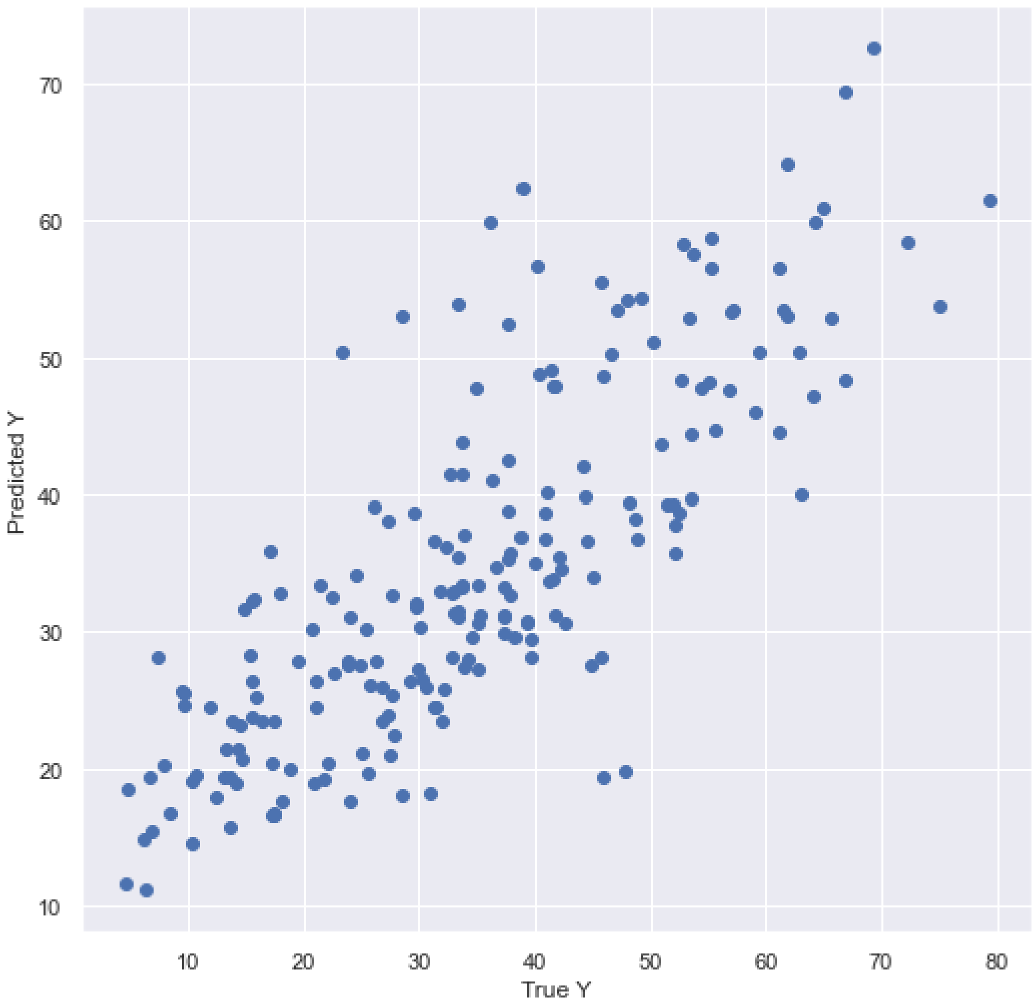
```
  0.02248514  0.11520355]
Mean squared error: 95.62
Coefficient of determination: 0.64
```

In [27]:

```
plt.scatter(Y_test,Y_test_pred)
plt.xlabel('True Y')
plt.ylabel('Predicted Y')
```

Out[27]:

```
Text(0, 0.5, 'Predicted Y')
```



In [ ]: