In [27]:

```
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
from sklearn.linear_model import LinearRegression
from sklearn.datasets import make_regression
%matplotlib notebook
```

In [2]:

```
X, y = make_regression()
```

In [6]:

```
print(X.shape, y.shape)
```

```
(100, 100) (100,)
```

In [18]:

```
X, y = make_regression(n_samples=1000, n_features=2, n_informative=2, noise=10 ,random_s
tate=1)
```

In [19]:

```
print(X.shape, y.shape)
```

```
(1000, 2) (1000,)
```

In [20]:

```
dfX = pd.DataFrame(X, columns=['f1', 'f2'])
dfX
```

Out[20]:

	f1	f2
0	**-0.915424**	**-1.945047**
1	**-2.304908**	**-0.592461**
2	**-0.437509**	**-0.781912**
3	**-1.061797**	**-0.569149**
4	**0.008359**	**0.034066**
...	**...**	**...**
995	**-0.479185**	**-0.418938**
996	**-1.899463**	**-1.168961**
997	**-0.468674**	**0.021863**
998	**-0.442655**	**0.557533**
999	**-0.139712**	**0.853282**

1000 rows × 2 columns

In [21]:

```
dfY = pd.DataFrame(y, columns=['y'])
dfY
```

Out[21]:

	y

	y
0	-198.367142
1	-71.877706
2	-86.970324
3	-69.649240
4	18.274576
...	...
995	-38.361458
996	-146.401642
997	-19.200111
998	47.016140
999	70.677107

1000 rows × 1 columns

In [22]:

```
print(type(X[:, 0]))
print(type(y))

print(X[:, 0].shapea)
print(y.shape)
```

```
<class 'numpy.ndarray'>
<class 'numpy.ndarray'>
(1000,)
(1000,)
```

In [23]:

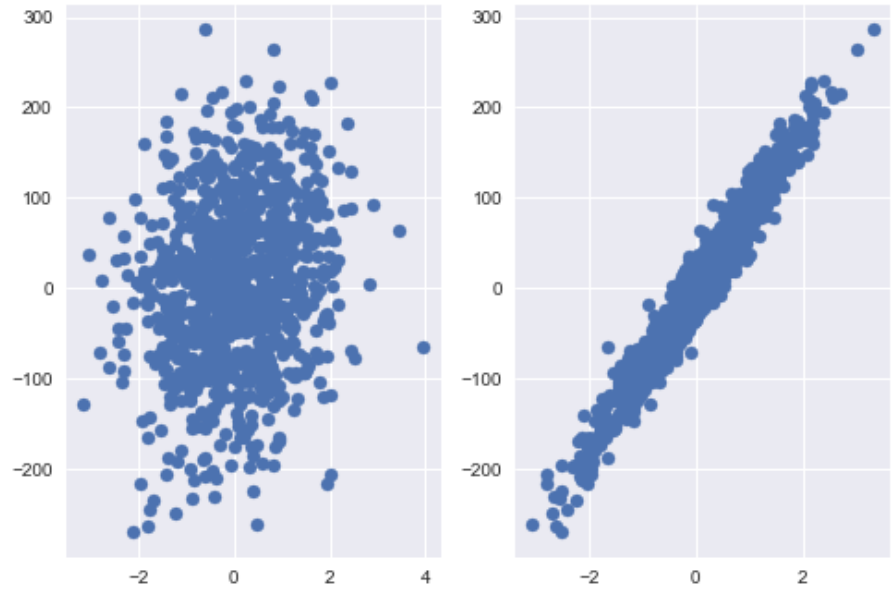
```
plt.style.use('seaborn')
```

In [24]:

```
plt.subplot(1, 2, 1)
plt.scatter(X[:, 0],a y)

plt.subplot(1, 2, 2)
plt.scatter(X[:, 1], y)

plt.show()
```

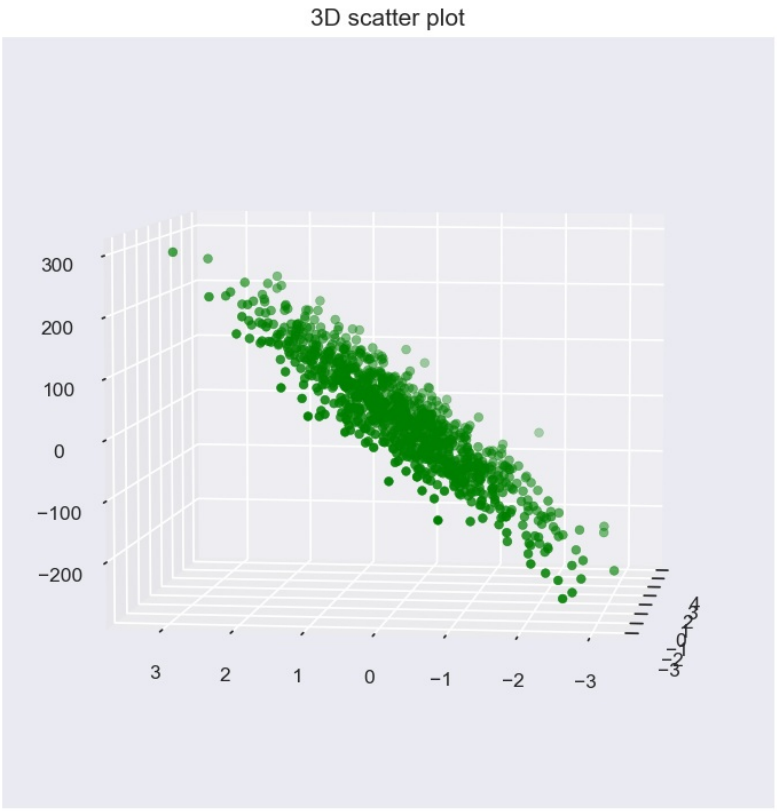


In [28]:

```
from mpl_toolkits import mplot3d
```

```
fig = plt.figure(figsize = (10, 7))
ax = plt.axes(projection='3d')

# Creating Plot
ax.scatter3D(X[:, 0], X[:, 1], y, color='green')
plt.title('3D scatter plot')
plt.show()
```



Linear Regression

In [29]:

```
model = LinearRegression()
```

In [30]:

```
model.fit(X, y)
```

Out[30]:

```
LinearRegression()
```

In [31]:

```
model.coef_
```

Out[31]:

```
array([16.62437639, 87.99042847])
```

In [33]:

```
model.intercept_
```

Out[33]:

-0.21766413760018422

In [34]:

```
print(type(X[0]))
print(X[0].shape)
```

```
<class 'numpy.ndarray'>
(2,)
```

In [37]:

```
model.predict([X[0], X[1]])
```

Out[37]:

```
array([-186.58153868,  -90.66624357])
```

In [38]:

```
y[0], y[1]
```

Out[38]:

```
(-198.36714169030483, -71.87770558832477)
```

In [41]:

```
# R2 score
model.score(X, y)
```

Out[41]:

```
0.988401838483157
```

In [42]:

```
# model.score?
```

Exploring

In [43]:

```
X, y = make_regression(n_samples=1000, n_features=3, n_informative=3, noise=10 ,random_s
tate=1)
```

In [44]:

```
print(X.shape, y.shape)
```

```
(1000, 3) (1000,)
```

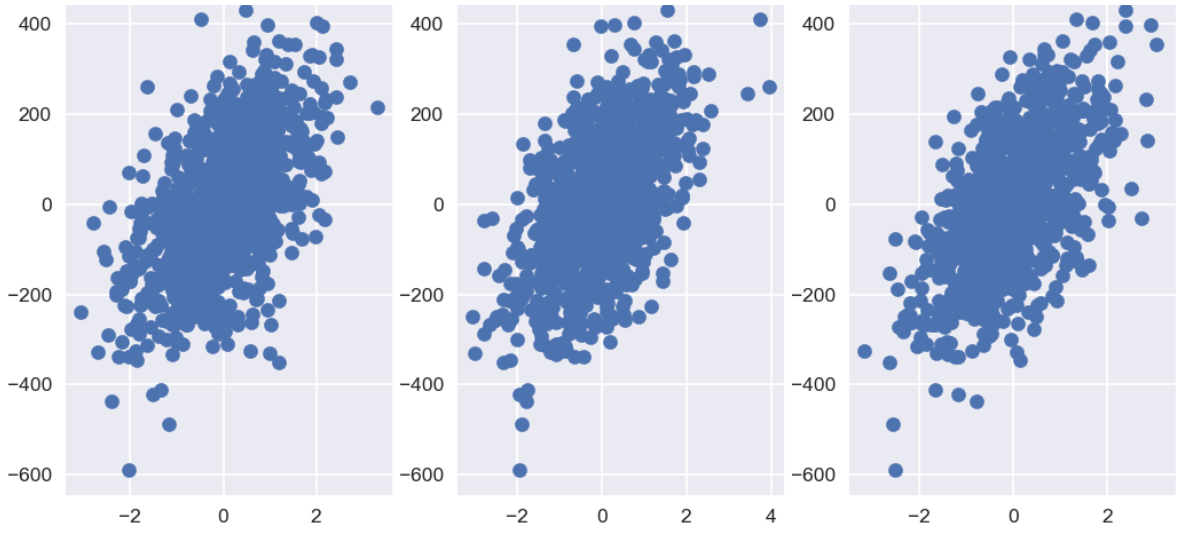
In [48]:

```
fig = plt.figure(figsize = (10, 5))
plt.subplot(1, 3, 1)
plt.scatter(X[:, 0], y)

plt.subplot(1, 3, 2)
plt.scatter(X[:, 1], y)

plt.subplot(1, 3, 3)
plt.scatter(X[:, 2], y)

plt.show()
```



In [ ]:

```
from mpl_toolkits import mplot3d

fig = plt.figure(figsize = (10, 7))
ax = plt.axes(projection='3d')

# Creating Plot
ax.scatter3D(X[:, 0], X[:, 1], y, color='green')
plt.title('3D scatter plot')
plt.show()
```

In [ ]:

```

```

In [49]:

```
model = LinearRegression()
model.fit(X, y)
print(model.coef_)
print(model.intercept_)

# R2 score
model.score(X, y)
```

```
[86.88487003 89.16247282 95.80681858]
-0.02574715594392263
```

Out[49]:

```
0.9957566230931877
```

In [ ]:

```

```

In [ ]:

```

```

In [ ]:

```

```

In [ ]:

```

```

In [ ]:

```

```

In [ ]:

In [ ]:

In [ ]:

In [ ]:

In [ ]: