```
import matplotlib.pyplot as plt
import numpy as np
from sklearn import linear_model
import random
```

```
def fun(x):
  res=x*20 +3
  error = res*random.uniform(-0.10,0.10)
  return res + error
```

```
val=[]
for i in range(0,200):
  g= random.uniform(1,500)
  h=fun(g)
  val.append((g,h))
  regression=linear_model.LinearRegression()
```

```
g,h=zip(*val)
print(g)
print(h)
maxofg=max(g)
minofg=min(g)
print(maxofg)
print(minofg)
```

```
(454.19179164719657, 47.03730527490814, 365.4908538655828, 284.8120440929466, 89.410385
(9410.362587914355, 962.0591003299375, 7176.7180713375365, 6144.4661492315245, 1789.113
497.35834754120026
1.2502103156962758
```

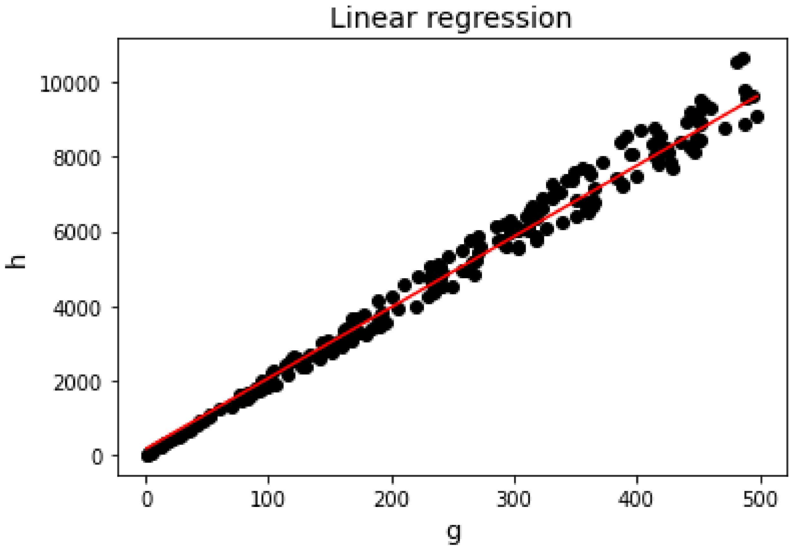
```
td_g = list(map(lambda g: [g], list(g[:-25])))
td_h=list(h[:-25])
td_g = list(map(lambda g: [g], list(g[-25:])))
td_h=list(h[-25:])
```

```
regression.fit(td_g,td_h)
s=regression.coef_[0]
t = regression.intercept_
print('h= {0}* g + {1}'.format(s,t))
```

```
h= 18.913978235680744* g + 199.71982644882246
```

```
plt.scatter(g, h, color='black')
plt.plot([minofg, maxofg], [t, s*maxofg + t], 'r')
plt.title('Linear regression', fontsize=14)
plt.xlabel('g', fontsize=13)
plt.ylabel('h', fontsize=13)
```

```
Text(0, 0.5, 'h')
```



```
print("Mean squared error is : %.4f" % np.mean((regression.predict(td_g) - td_h) ** 2))
```

```
Mean squared error is : 96873.4607
```

```
print('Variance is : %.4f' % regression.score(td_g, td_h))
```

```
Variance is : 0.9836
```

Colab paid products - Cancel contracts here

✓ 0s completed at 00:58 ● ✕