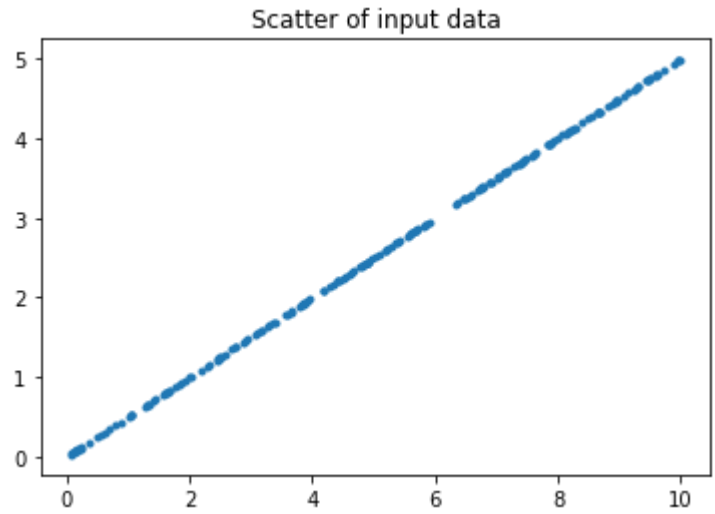
1.a) Simplest case

In [1]:

```
import numpy as np
import matplotlib.pyplot as plt

%matplotlib inline
```

In [11]:

```
#Generate errorless data

slope = 0.5 # can be randomised with slope = np.random()
X = np.random.rand(200)*10
Y = slope*X

plt.scatter(X,Y,marker=".")
plt.title("Scatter of generated data")
plt.show()
```



In [3]:

```
# calculate error for w1 [-10,10]

mse = []

for w1 in np.arange(-10,10,0.5):
    Y_ = w1*X
    mean_squared_error = np.mean((Y-Y_)**2)
    mse.append(mean_squared_error)
```

In [12]:

```
plt.plot(np.arange(-10,10,0.5),mse,color="r")
plt.title("Error vs w")
plt.show()
```

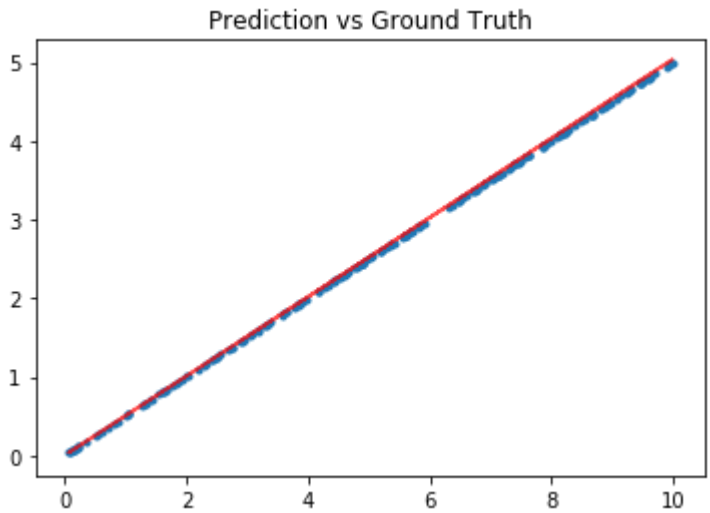
In [13]:

```
w = 0
mean_squared_error = 1
learning_rate = 0.01

while mean_squared_error > 0.001:
    Y_ = w*X
    mean_squared_error = np.mean((Y-Y_)**2)
    dL_dw = np.mean(2*(Y-Y_)*X)
    w += learning_rate*dL_dw

print("w : "+str(w))
```

```
W : 0.498709513098
```

In [23]:

```
plt.plot(X,Y_,color="r",alpha=0.7)
plt.scatter(X,Y,marker=".")
plt.title("Prediction vs Ground Truth")
plt.show()
```



Now with error

In [24]:

```
# Introduce error into data

Y_error = Y + np.random.normal(0,0.5,200)
plt.scatter(X,Y_error,marker=".")
plt.title("Generted data with error")
plt.show()
```

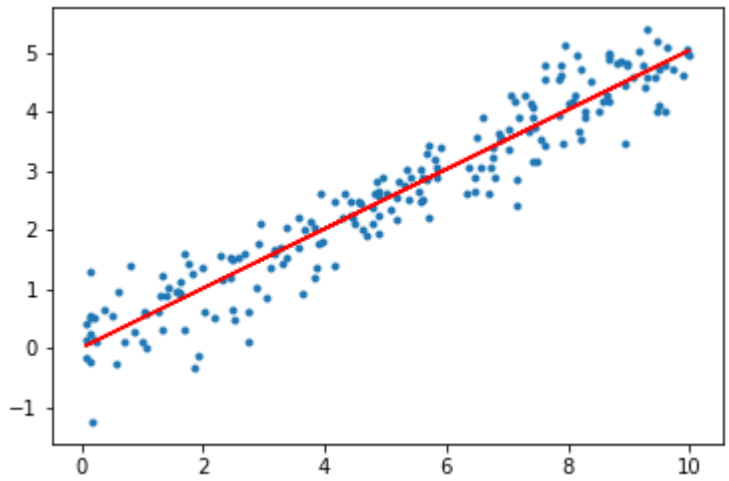
In [25]:

```
w = 0
mean_squared_error = 1
learning_rate = 0.005

for _ in range(2000):
    Y_ = w*X
    mean_squared_error = np.mean((Y_error-Y_)**2)
    dL_dw = np.mean(2*(Y_error-Y_)*X)
    w += learning_rate*dL_dw

print("w : " + str(w))
```

```
w : 0.49681496688
```

In [18]:

```
plt.plot(X,Y_,color="r")
plt.scatter(X,Y_error,marker=".")
plt.title("Prediction vs Ground Truth")
plt.show()
```



In [20]:

```
# generate test data and find performance

X_test = np.random.rand(200)*10
Y_test = slope*X + np.random.normal(0,0.5,200)
test_error = np.mean((Y_test-w*X_test)**2)
print("test_error : " + str(test_error))
```

```
test_error : 4.79462279212
```