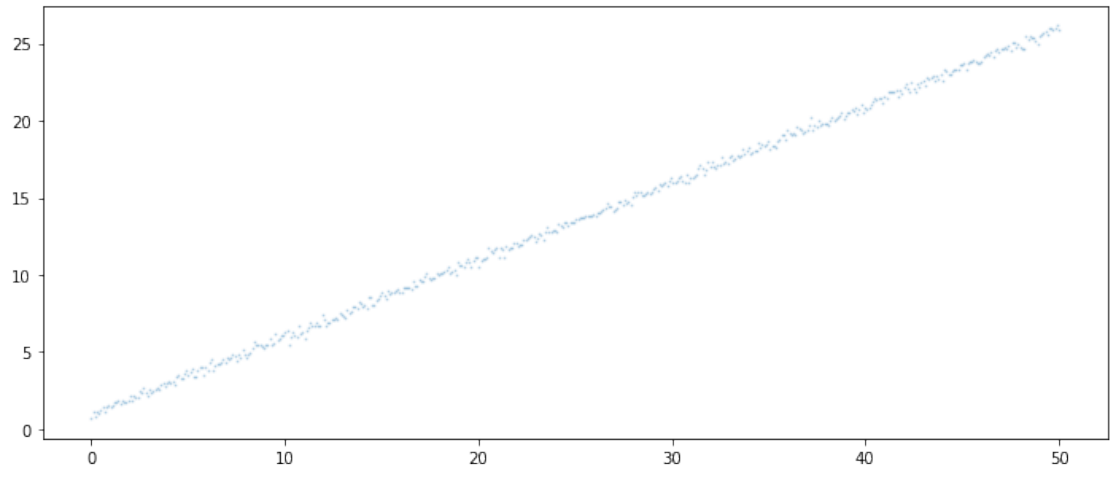
Task3

November 10, 2017

In [47]:
```
import numpy as np
import scipy
from scipy import stats, optimize
import matplotlib.pyplot as plt
%matplotlib inline
```

In [63]:
```
N = 500
x_max = 50
x = np.linspace(0, x_max, N)
y = 0.5 * x + 1 + scipy.stats.norm(0, 0.2).rvs(size=N)
```

In [66]:
```
plt.figure(figsize=(12, 5))
plt.scatter(x, y, s=1, alpha=0.2)
plt.show()
```

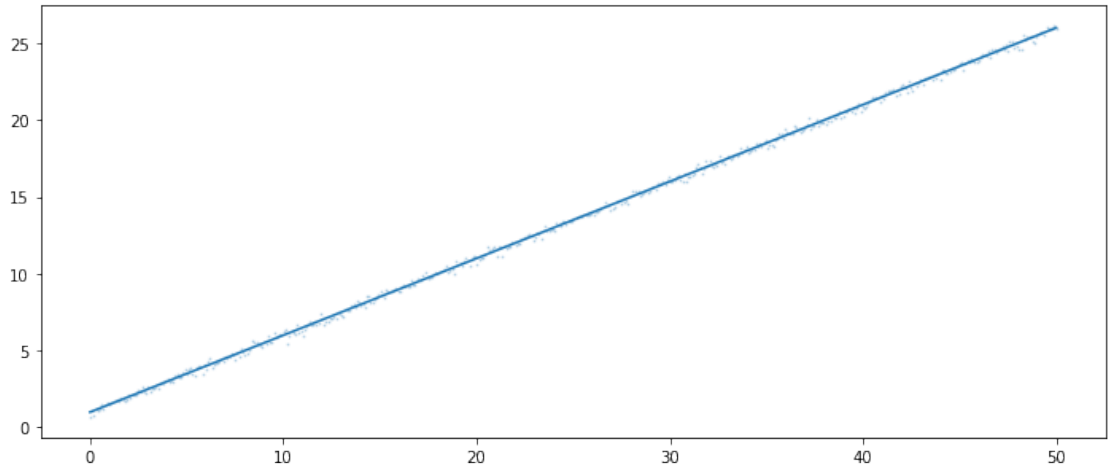


In [67]:
```
def f_mse(args):
    y_pred = args[0] * x + args[1]
    return np.sqrt(np.sum((y - y_pred)**2)) / N
```

In [68]:
```
k, b = optimize.minimize(f_mse, [0., 0.]).x
print ("predicted values:\n", "k =", k, "\nb =", b)
```

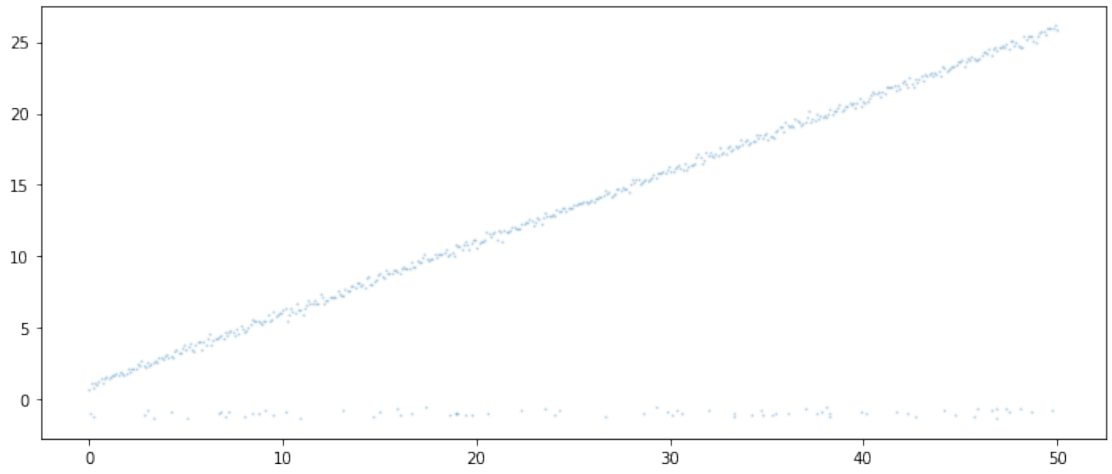
```
        y_pred = k * x + b
        plt.figure(figsize=(12, 5))
        plt.plot(x, y_pred)
        plt.scatter(x, y, s=1, alpha=0.2)
        plt.show()
```

```
predicted values:
 k = 0.500274903425
b = 0.987247798137
```



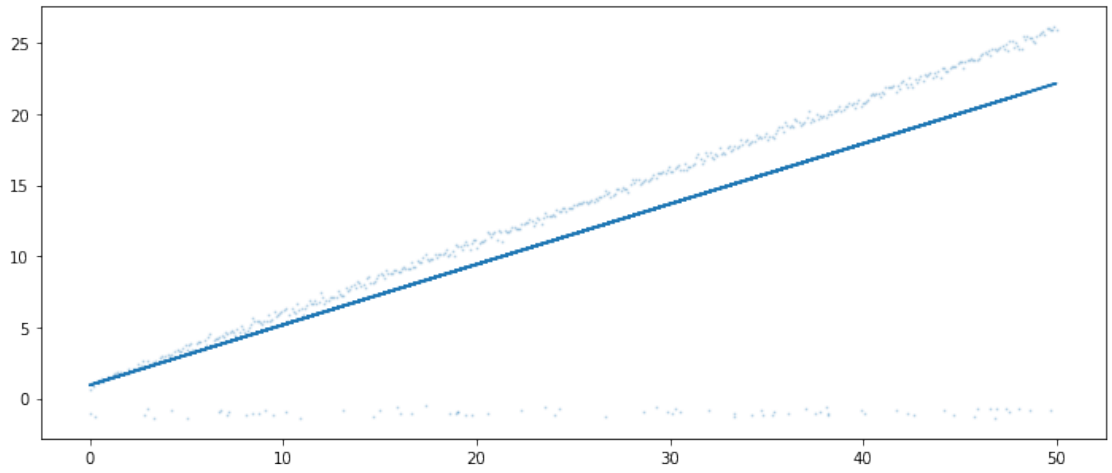
In [69]:
```
N_noise = 75
x_noise = stats.uniform(0, x_max).rvs(size=N_noise)
y_noise = stats.norm(0, 0.2).rvs(size=N_noise) - 1
```

In [70]:
```
x = np.hstack((x, x_noise))
y = np.hstack((y, y_noise))
plt.figure(figsize=(12, 5))
plt.scatter(x, y, s=1, alpha=0.2)
plt.show()
```




```
In [71]: k, b = optimize.minimize(f_mse, [0., 0.]).x
         print ("predicted values:\n", "k =", k, "\nb =", b)
         y_pred = k * x + b
         plt.figure(figsize=(12, 5))
         plt.plot(x, y_pred)
         plt.scatter(x, y, s=1, alpha=0.2)
         plt.show()
```

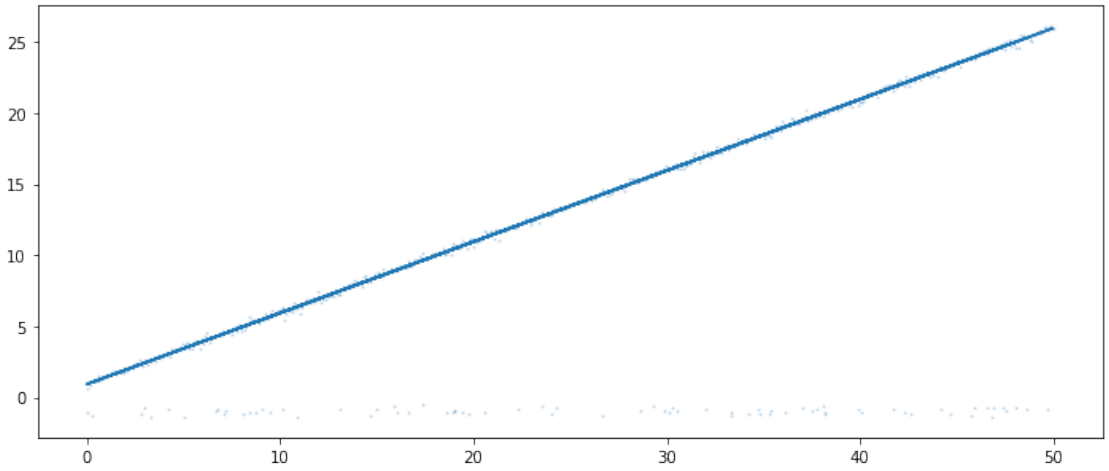
```
predicted values:
 k = 0.423847247202
b = 0.950069515843
```



```
In [72]: def f_mae(args):
             y_pred = args[0] * x + args[1]
             return np.sum(np.abs(y - y_pred)) / N

In [73]: k, b = optimize.minimize(f_mae, [0., 0.]).x
         print ("predicted values:\n", "k =", k, "\nb =", b)
         y_pred = k * x + b
         plt.figure(figsize=(12, 5))
         plt.plot(x, y_pred)
         plt.scatter(x, y, s=1, alpha=0.2)
         plt.show()

predicted values:
 k = 0.500445674583
b = 0.940162624375
```



1 :

, MAE, , MSE . , , MAE, , .