In [1]:

```
import numpy as np
from sklearn.linear_model import LinearRegression
from sklearn.metrics import mean_squared_error, r2_score
import matplotlib.pyplot as plt
import pandas as pd
plt.style.use('seaborn')

df = pd.read_csv('data/walk.csv')
print(df.head())

y_train = df['distance']
x_train = df[['time','temperature','age','weight']]

model = LinearRegression()
model.fit(x_train,y_train)
# Y = b1X1 + b2X2 + ... + a
b = model.coef_
a = model.intercept_

#Y=1.8(time) - 0.80(temperature) - 0.21(age) - 0.52(weight) = 41.98
y_pred = model.predict(x_train)
r2 = r2_score(y_train,y_pred)
mse = mean_squared_error(y_train,y_pred)
print('R2: %.2f'%(r2))
print('MSE: %.2f'%(mse))

print("Musltiple liner regression equation:")
print("Y = %.2f(Time) + %.2f(Temp) + %.2f(Age)  + %.2f(weight) = %.2f "%(b[0],b[1],b[2],b[3],a))
```

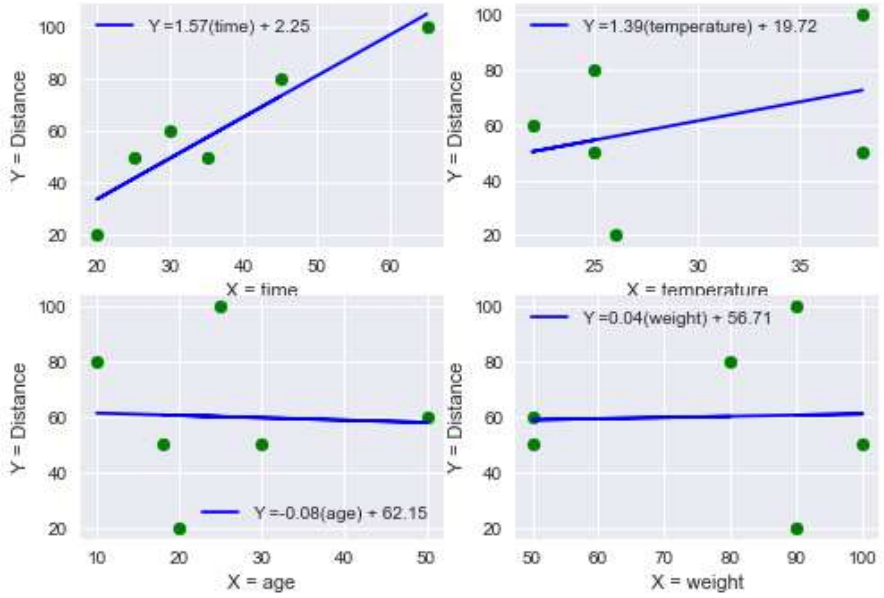
```
   distance  time  temperature  age  weight
0       100    65           38   25      90
1        50    35           38   18     100
2        20    20           26   20      90
3        50    25           25   30      50
4        60    30           22   50      50
R2: 0.98
MSE: 12.07
Musltiple liner regression equation:
Y = 1.81(Time) + -0.09(Temp) + -0.21(Age)  + -0.52(weight) = 41.98
```

In [2]:

```
import numpy as np
from sklearn.linear_model import LinearRegression
from sklearn.metrics import mean_squared_error, r2_score
import matplotlib.pyplot as plt
import pandas as pd
plt.style.use('seaborn')

df = pd.read_csv('data/walk.csv')
print(df.head())

def caculator(data,plt,subplot):
    y_train = df['distance']
    x_train = df[[data]]

    model = LinearRegression()
    model.fit(x_train,y_train)
    # Y = b1X1 + b2X2 + ... + a
    b = model.coef_
    a = model.intercept_

    y_pred = model.predict(x_train)
    r2 = r2_score(y_train,y_pred)
    mse = mean_squared_error(y_train,y_pred)
    print("Linea Equation x0\t: Y = %.2f(%s)+%.2f"%(b,data,a))
    print('R2: %.2f'%(r2))
    print('MSE: %.2f'%(mse))
```

```
    plt.subplot(subplot)
    plt.scatter(x_train,y_train,color ="green")
    plt.plot(x_train,y_pred,color = "blue",label="Y =%.2f(%s) + %.2f"%(b,data,a))
    plt.xlabel('X = %s'%(data))
    plt.ylabel('Y = Distance')
    plt.legend()


subplot=221
xno=0
x_data = ['time', 'temperature', 'age', 'weight']
for i in range(len(x_data)):
    caculator(x_data[i],plt,subplot)
    subplot=subplot+1
    xno=xno+1

plt.show()
```

```
   distance  time  temperature  age  weight
0       100    65           38   25      90
1        50    35           38   18     100
2        20    20           26   20      90
3        50    25           25   30      50
4        60    30           22   50      50
Linea Equation x0       : Y = 1.57(time)+2.25
R2: 0.87
MSE: 82.08
Linea Equation x0       : Y = 1.39(temperature)+19.72
R2: 0.13
MSE: 552.31
Linea Equation x0       : Y = -0.08(age)+62.15
R2: 0.00
MSE: 632.21
Linea Equation x0       : Y = 0.04(weight)+56.71
R2: 0.00
MSE: 632.62
```



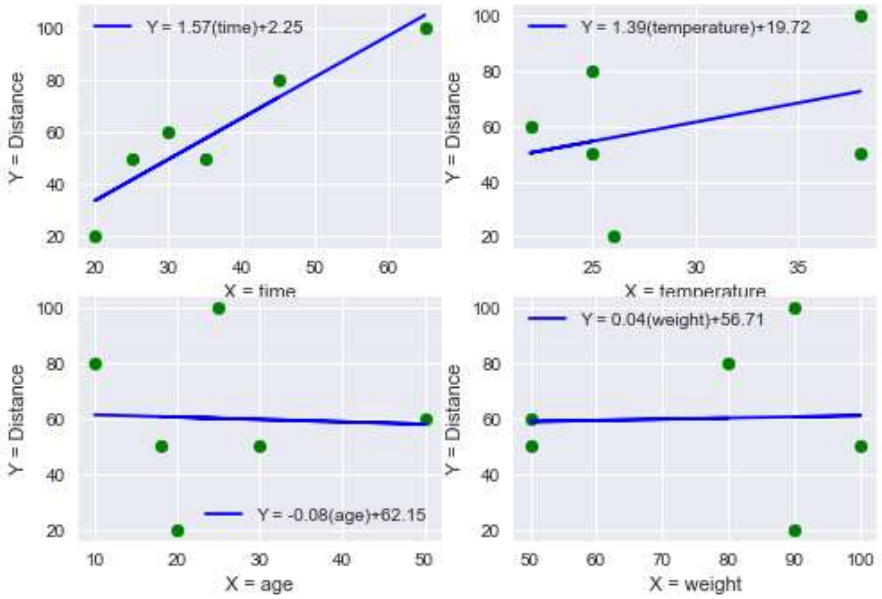
In [3]:

```
import numpy as np
from sklearn.linear_model import LinearRegression
from sklearn.metrics import mean_squared_error, r2_score
import pandas as pd
import matplotlib.pyplot as plt
np.set_printoptions(precision=2)
plt.style.use('seaborn')

df = pd.read_csv('data/walk.csv')
```

```
print(df.head())

x_train = df[['time', 'temperature', 'age', 'weight']]
y_train = df['distance']

model = LinearRegression()
model.fit(x_train, y_train)

b = model.coef_
a = model.intercept_

subplot=221
xno=0

y_pred = model.predict(x_train)
r2 = r2_score(y_train, y_pred)
MSE = mean_squared_error(y_train, y_pred)
print("R2\t:",r2)
print("MSE\t:",MSE)
print('Predicted response of X:',y_pred)


print('\n\nLiner Regression of each single X+++')
x_data = ['time', 'temperature', 'age', 'weight']
for i in range(len(x_data)):

  x_train = df[[x_data[i]]]
  model = LinearRegression()
  model.fit(x_train, y_train)
  b = model.coef_
  a = model.intercept_
  y_pred = model.predict(x_train)
  r2 = r2_score(y_train, y_pred)
  MSE = mean_squared_error(y_train, y_pred)
  print("Linea Equation x0\t: Y = %.2f(%s)+%.2f"%(b,x_data[i],a))
  print("R2_x%d\t: %.2f"%(xno,r2))
  print("MSE_x%d\t: %.2f"%(xno,MSE))



  plt.subplot(subplot)
  plt.scatter(x_train, y_train, color='green')
  plt.plot(x_train,y_pred, color='blue', label = "Y = %.2f(%s)+%.2f"%(b,x_data[i],a))
  plt.xlabel('X = %s'%(x_data[i]))
  plt.ylabel('Y = Distance')
  plt.legend()

  subplot=subplot+1
  xno=xno+1

plt.show()
```

```
   distance  time  temperature  age  weight
0       100    65           38   25      90
1        50    35           38   18     100
2        20    20           26   20      90
3        50    25           25   30      50
4        60    30           22   50      50
R2      : 0.9809362360702043
MSE     : 12.073717155537262
Predicted response of X: [103.5   45.55  24.33  52.31  57.34  76.97]


Liner Regression of each single X+++
Linea Equation x0       : Y = 1.57(time)+2.25
R2_x0   : 0.87
MSE_x0  : 82.08
Linea Equation x0       : Y = 1.39(temperature)+19.72
R2_x1   : 0.13
```

```
MSE_x1  : 552.31
Linea Equation x0       : Y = -0.08(age)+62.15
R2_x2   : 0.00
MSE_x2  : 632.21
Linea Equation x0       : Y = 0.04(weight)+56.71
R2_x3   : 0.00
MSE_x3  : 632.62
```

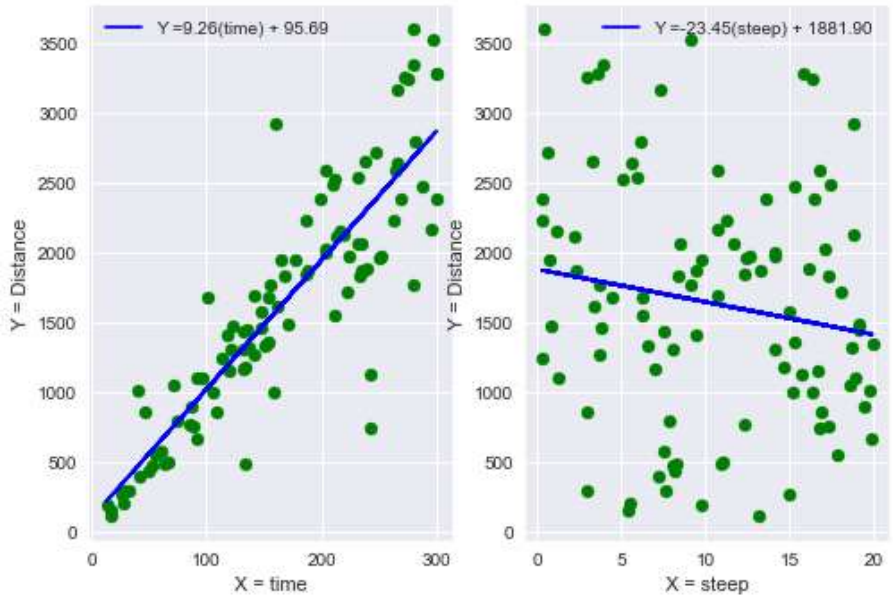


In [4]:

```
import numpy as np
from sklearn.linear_model import LinearRegression
from sklearn.metrics import mean_squared_error, r2_score
import matplotlib.pyplot as plt
import pandas as pd
plt.style.use('seaborn')

df = pd.read_csv('data/multi_run.csv')
print(df.head())

def caculator(data,plt,subplot):
    y_train = df['distance']
    x_train = df[[data]]

    model = LinearRegression()
    model.fit(x_train,y_train)
    # Y = b1X1 + b2X2 + ... + a
    b = model.coef_
    a = model.intercept_

    y_pred = model.predict(x_train)
    r2 = r2_score(y_train,y_pred)
    mse = mean_squared_error(y_train,y_pred)
    print("Linea Equation x0\t: Y = %.2f(%s)+%.2f"%(b,data,a))
    print('R2: %.2f'%(r2))
    print('MSE: %.2f'%(mse))

    plt.subplot(subplot)
    plt.scatter(x_train,y_train,color ="green")
    plt.plot(x_train,y_pred,color = "blue",label="Y =%.2f(%s) + %.2f"%(b,data,a))
    plt.xlabel('X = %s'%(data))
    plt.ylabel('Y = Distance')
    plt.legend()

y_train = df['distance']
x_train = df[['time','steep']]

model = LinearRegression()
```

```
model.fit(x_train,y_train)
# Y = b1X1 + b2X2 + ... + a
b = model.coef_
a = model.intercept_
y_pred = model.predict(x_train)
r2 = r2_score(y_train,y_pred)
mse = mean_squared_error(y_train,y_pred)


print("\nMultiple linear regression equation:\tY = %.2f(time) + %.2f(steep) + %.2f "%(b[0],b[1],a))
print('R2: %.2f'%(r2))
print('MSE: %.2f\n'%(mse))

x_new = np.array([[10,20],[15,2],[20,10]])
y_pred_new = model.predict(x_new)
print("Predicted response of X:")
print("10,20\t%.2f"%(y_pred_new[0]))
print("15,2\t%.2f"%(y_pred_new[1]))
print("20,10\t%.2f\n"%(y_pred_new[2]))

sp=121
xno=0
data = ['time','steep']
for i in range(len(data)):
    caculator(data[i],plt,sp)
    sp=sp+1
    xno=xno+1
plt.show()
```

```
   distance  time  steep
0    109.58    17  13.21
1    153.77    17   5.41
2    267.12    26  14.96
3    199.50    27   5.50
4    297.04    30   2.96

Multiple linear regression equation:    Y = 9.19(time) + -16.06(steep) + 274.36
R2: 0.78
MSE: 161332.22

Predicted response of X:
10,20   45.07
15,2    380.12
20,10   297.59

Linea Equation x0       : Y = 9.26(time)+95.69
R2: 0.77
MSE: 170443.57
Linea Equation x0       : Y = -23.45(steep)+1881.90
R2: 0.03
MSE: 723421.19
```


Y =9.26(time) + 95.69
Y = Distance
X = time
Y =-23.45(steep) + 1881.90
Y = Distance
X = steep

In [ ]:

In [ ]:

In [ ]: