Table of Contents

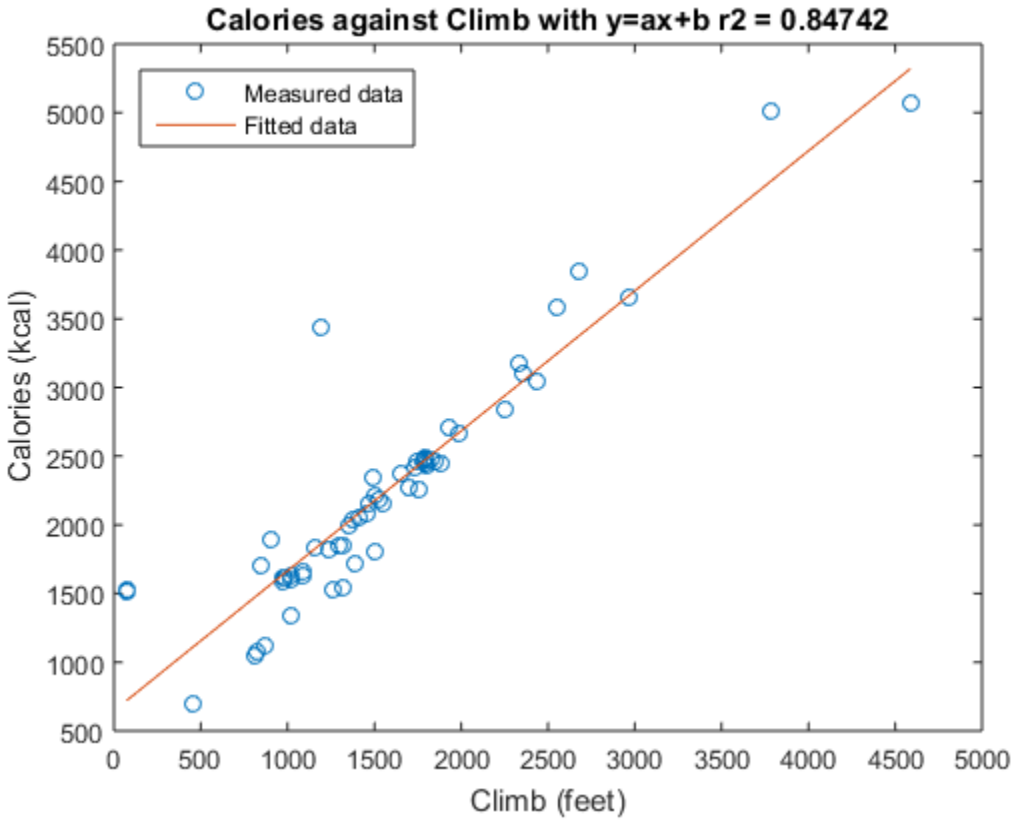

```
clc,clear;
GetRideData;
```

Part 1

```
disp('Part 1.................................Climb is independent')
xclimb = climb;         %assigning climb to xclimb
y1Calories = calories;      %assigning climb to y1Calories

% Computing the mean and variance of Climb,Calories
Mean_of_Climb = mean(climb);
Variance_of_Climb = var(climb);
disp(['Mean of Climb is ',num2str(Mean_of_Climb)])

disp(['Variance of Climb is ',num2str(Variance_of_Climb)])

disp('-----------------------------------------------------------------')
Mean_of_Calories = mean(calories);
Variance_of_Calories = var(calories);

disp(['Mean of Calories is ',num2str(Mean_of_Calories)])

disp(['Variance of Calories is ',num2str(Variance_of_Calories)])

%Computing the first order value of a, b fitting the data of climb and
%calories

p=polyfit(xclimb,y1Calories,1);

a=p(1);
b=p(2);
disp(['a is ', num2str(a)]);
disp(['b is ', num2str(b)]);
yLfit1E1=a*xclimb+b;
%Computing the correllation coefficient of fitted equation to make
%decision, the equation with higher r will be chosen.
n=numel(y1Calories);        %number of elements in data
Mean1E1 = sum(y1Calories)/n;

S1E1 = sum((y1Calories-Mean1E1).^2);

A1E1= sum((yLfit1E1-y1Calories).^2);
```

```
r21E1 = 1-A1E1/S1E1;                %correlation coeficient of y=ab+c


% Computing the first order value of C, d fitting the data of climb
 and
%calories
%The Equation y=C*exp^dx is first lineerized by taking natural log of
 both
%sides

pp=polyfit(xclimb,log(y1Calories),1);           %fitting for Part 1
 Equarion 2
d=pp(1);
C=exp(pp(2));

disp(['C is ', num2str(C)]);
disp(['d is ', num2str(d)]);
yLfitE12=C*exp(d*xclimb);

Mean1E2 = sum(y1Calories)/n;

S1E2 = sum((y1Calories-Mean1E2).^2);

A1E2= sum((yLfitE12-y1Calories).^2);

r21E2 = 1-A1E2/S1E2;        %correlation coeficient of y=Cepx(dx)


figure(1)
plot(xclimb,y1Calories,'o',xclimb,yLfit1E1)
legend('Measured data','Fitted data','Location','northwest')
title(['Calories against Climb with y=ax+b', ' r2 = ',
 num2str(r21E1)])
xlabel('Climb (feet)')
ylabel('Calories (kcal)')
hold on

%At Climbing feet of 6500
newCfeet = 6500;
newCal = a*newCfeet+b;

disp('**********************************************************************')
disp(['At 6500 Climbing feet,', num2str(newCal), ' kcal is the
 Calories predicted by the choosen model: y=ax+b'])

Part 1.................................Climb is independent
Mean of Climb is 1573.0167
Variance of Climb is 562794.3556
----------------------------------------------------------------
Mean of Calories is 2249.15
Variance of Calories is 689450.9432
a is 1.0189
b is 646.4203
```

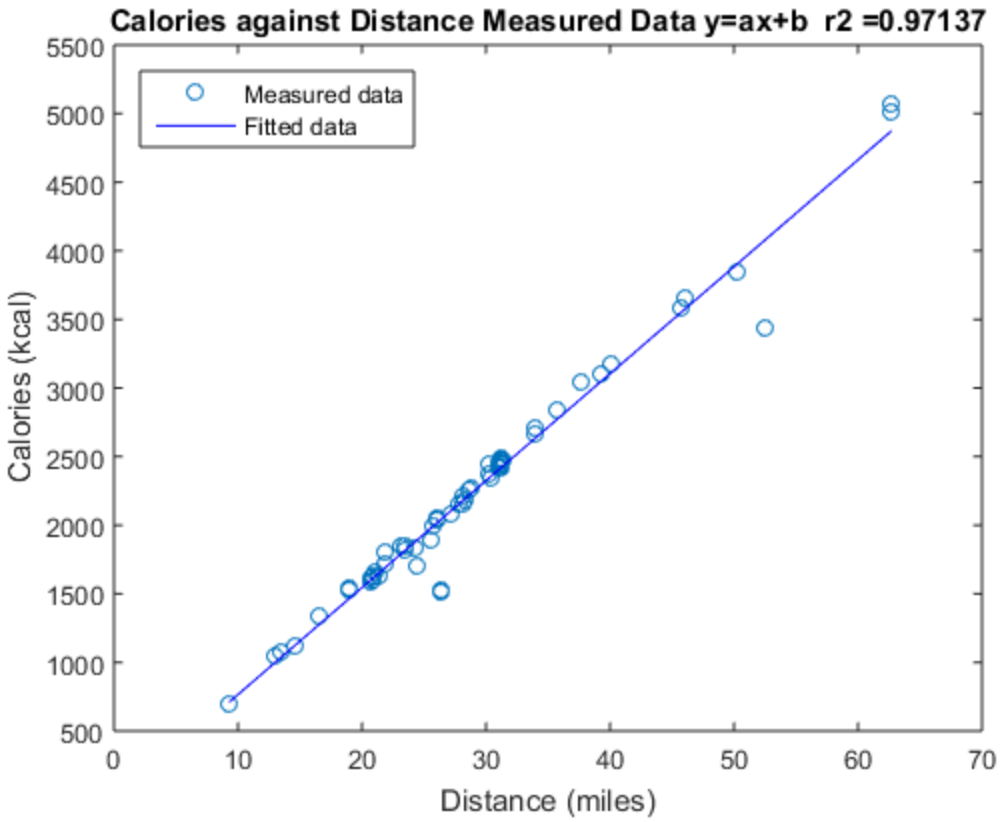
```
C is 1105.7362
d is 0.00041186
**********************************************************************
At 6500 Climbing feet,7269.2 kcal is the Calories predicted by the
 choosen model: y=ax+b
```



Part 2

```
disp('**********************************************************************')
disp('**********************************************************************')
disp('Part 2.................................Distance is
 independent')
%Declearation
xdistance=distance;                 %Assigning distance to xdistance
y2calories=calories;                %Assigning calories to y2calories

pL=polyfit(xdistance,y2calories,1);         %fitting for y=ax+b
a2=pL(1);                                 %collecting constant a into a2
b2=pL(2);                                  %collecting constant b into
 b2

yLfit2E1calories=a2*xdistance+b2;           %fitted data

%Displaying a and b
disp('For Part 2')
    disp(['a is ', num2str(a2)]);
```

```
disp(['b is ', num2str(b2)]);

% Equation 2, y=Gx^f was first linearized by taking natural log of
 both
% sides
p3=polyfit(log(xdistance),log(y2calories),1);          %first order
 fitting for log(y)=f*logx+log(G)
f=p3(1);                                %collecting f
G=exp(p3(2));                           %collecting G
yLfit2E2 = G*(distance.^f);             %fitted data

disp('-----------------------------------------------------------------')
disp('For Part 2')
    disp(['G is ', num2str(G)]);
disp(['f is ', num2str(f)]);
% Computing Correlation coefficient r22E1 for part 2 equation 1
n=numel(y2calories);
Mean2E1 = sum(y2calories)/n;

S2E1 = sum((y2calories-Mean2E1).^2);

A2E1= sum((yLfit2E1calories-y2calories).^2);

r22E1 = 1-A2E1/S2E1;

% Computing Correlation coefficient r22E2 for Part 2 equation 2
Mean2E2 = sum(y2calories)/n;

S2E2 = sum((y2calories-Mean2E2).^2);

A2E2= sum((yLfit2E2-y2calories).^2);

r22E2 = 1-A2E2/S2E2;

figure(2)

plot(xdistance,y2calories,'o',xdistance,yLfit2E1calories,'b')
legend('Measured data','Fitted data','Location','northwest')
title(['Calories against Distance Measured Data y=ax+b  ','r2 =',
 num2str(r22E1)])
xlabel('Distance (miles)')
ylabel('Calories (kcal)')
hold on

**********************************************************************
**********************************************************************
Part 2.....................................Distance is independent
For Part 2
a is 77.945
b is -13.8536
-----------------------------------------------------------------
For Part 2
G is 78.1008
f is 0.99695
```

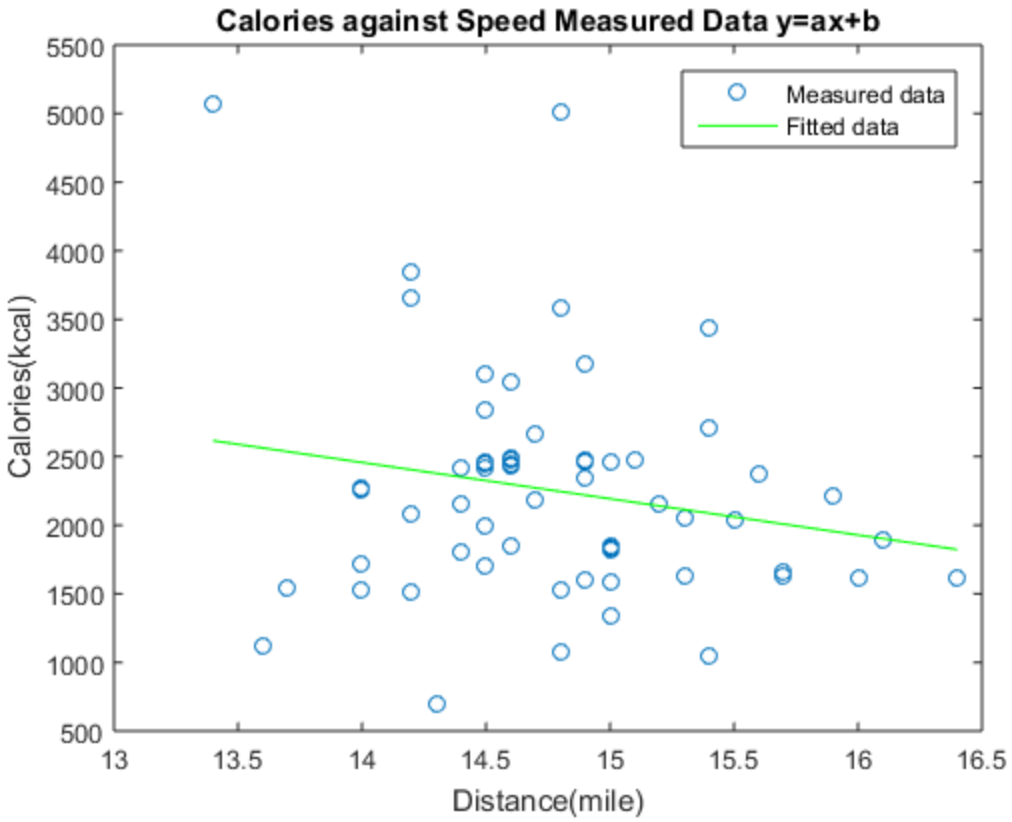


Part 3

```
disp('**************************************************************************')
disp('**************************************************************************')
disp('Part 3.................................Speed is independent')
%Declearation
xspeed=speed;                          %Allocating speed to xspeed
y3calories=calories;                   %Allocating calories to calories

pL3=polyfit(xspeed,y3calories,1);         %fitting for y=ax+b
a3=pL3(1);
b3=pL3(2);
yLfit3E1calories=a3*xspeed+b3;           %fitted data

%Displaying a and b
disp('For Part 3')
    disp(['a is ', num2str(a3)]);
disp(['b is ', num2str(b3)]);

Mean_of_Speed = mean(speed);
Variance_of_Speed = var(speed);
disp(['Mean of Speed is ',num2str(Mean_of_Speed)])

disp(['Variance of Speed is ',num2str(Variance_of_Speed)])
```

```
disp('-----------------------------------------------------------------')
Mean_of_Calories = mean(calories);
Variance_of_Calories = var(calories);

disp(['Mean of Calories is ',num2str(Mean_of_Calories)])

disp(['Variance of Calories is ',num2str(Variance_of_Calories)])

figure(3)
plot(xspeed,y3calories,'o',xspeed,yLfit3E1calories,'g')
legend('Measured data','Fitted data','Location','northeast')
title('Calories against Speed Measured Data y=ax+b')
xlabel('Distance(mile)')
ylabel('Calories(kcal)')
hold on
```

```
**********************************************************************
**********************************************************************
Part 3.................................Speed is independent
For Part 3
a is -264.3182
b is 6157.5344
Mean of Speed is 14.7867
Variance of Speed is 0.3727
-----------------------------------------------------------------
Mean of Calories is 2249.15
Variance of Calories is 689450.9432
```



Published with MATLAB® R2015a