AkashkumarPrasad Rollno:80

Lab 8: Supervised Learning - Regression

Theory:

Regression: Regression refers to a type of supervised machine learning technique that is used to predict any continuous-valued attribute. Regression helps any business organization to analyze the target variable and predictor variable relationships. It is a most significant tool to analyze the data that can be used for financial forecasting and time series modeling.

Linear Regression: Linear regression is the type of regression that forms a relationship between the target variable and one or more independent variables utilizing a straight line

Multiple Regression: Multiple regression analysis is a statistical technique that analyzes the relationship between two or more variables and uses the information to estimate the value of the dependent variables. In multiple regression, the objective is to develop a model that describes a dependent variable y to more than one independent variable.

Below is the sample data representing the observations –

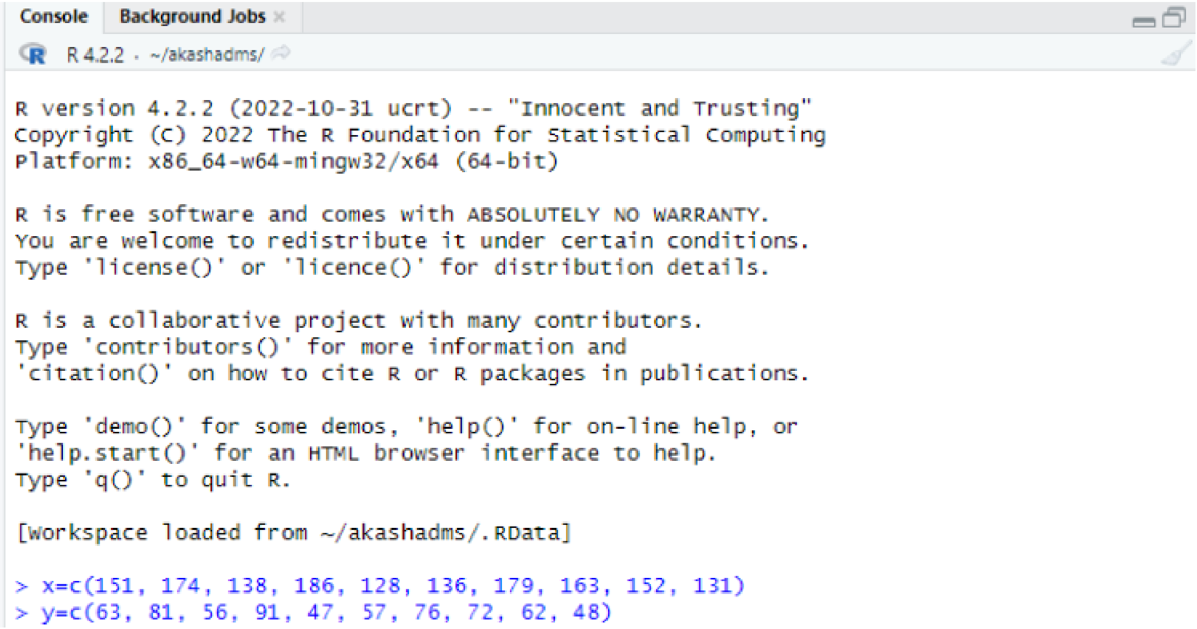
# Values of height

151, 174, 138, 186, 128, 136, 179, 163, 152, 131

# Values of weight.

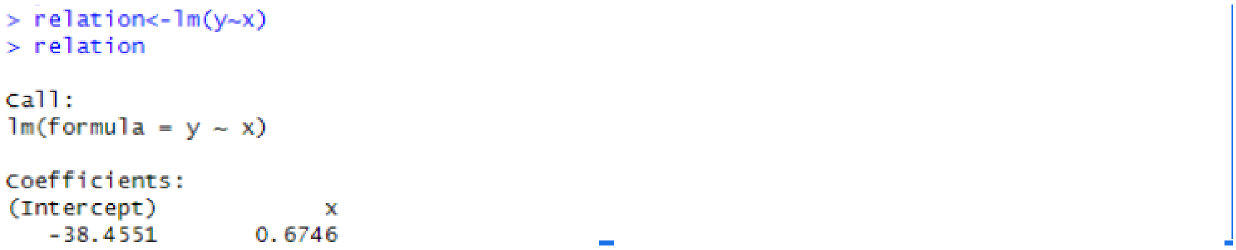
63, 81, 56, 91, 47, 57, 76, 72, 62, 48

a. Create height and weight vectors using the above values



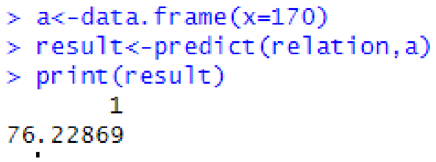
```
R version 4.2.2 (2022-10-31 ucrt) -- "Innocent and Trusting"
Copyright (C) 2022 The R Foundation for Statistical Computing
Platform: x86_64-w64-mingw32/x64 (64-bit)

R is free software and comes with ABSOLUTELY NO WARRANTY.
You are welcome to redistribute it under certain conditions.
Type 'license()' or 'licence()' for distribution details.

R is a collaborative project with many contributors.
Type 'contributors()' for more information and
'citation()' on how to cite R or R packages in publications.

Type 'demo()' for some demos, 'help()' for on-line help, or
'help.start()' for an HTML browser interface to help.
Type 'q()' to quit R.

[Workspace loaded from ~/akashadms/.RData]

> x=c(151, 174, 138, 186, 128, 136, 179, 163, 152, 131)
> y=c(63, 81, 56, 91, 47, 57, 76, 72, 62, 48)
```

b.Create relationship model &amp; get the coefficients using linear model function of R (lm).



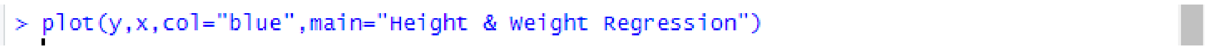
```
> relation<-lm(y~x)
> relation

Call:
lm(formula = y ~ x)

Coefficients:
(Intercept)            x
   -38.4551       0.6746
```

c. Get the summary of the relationship and predict the weight of new persons whose height is 170.



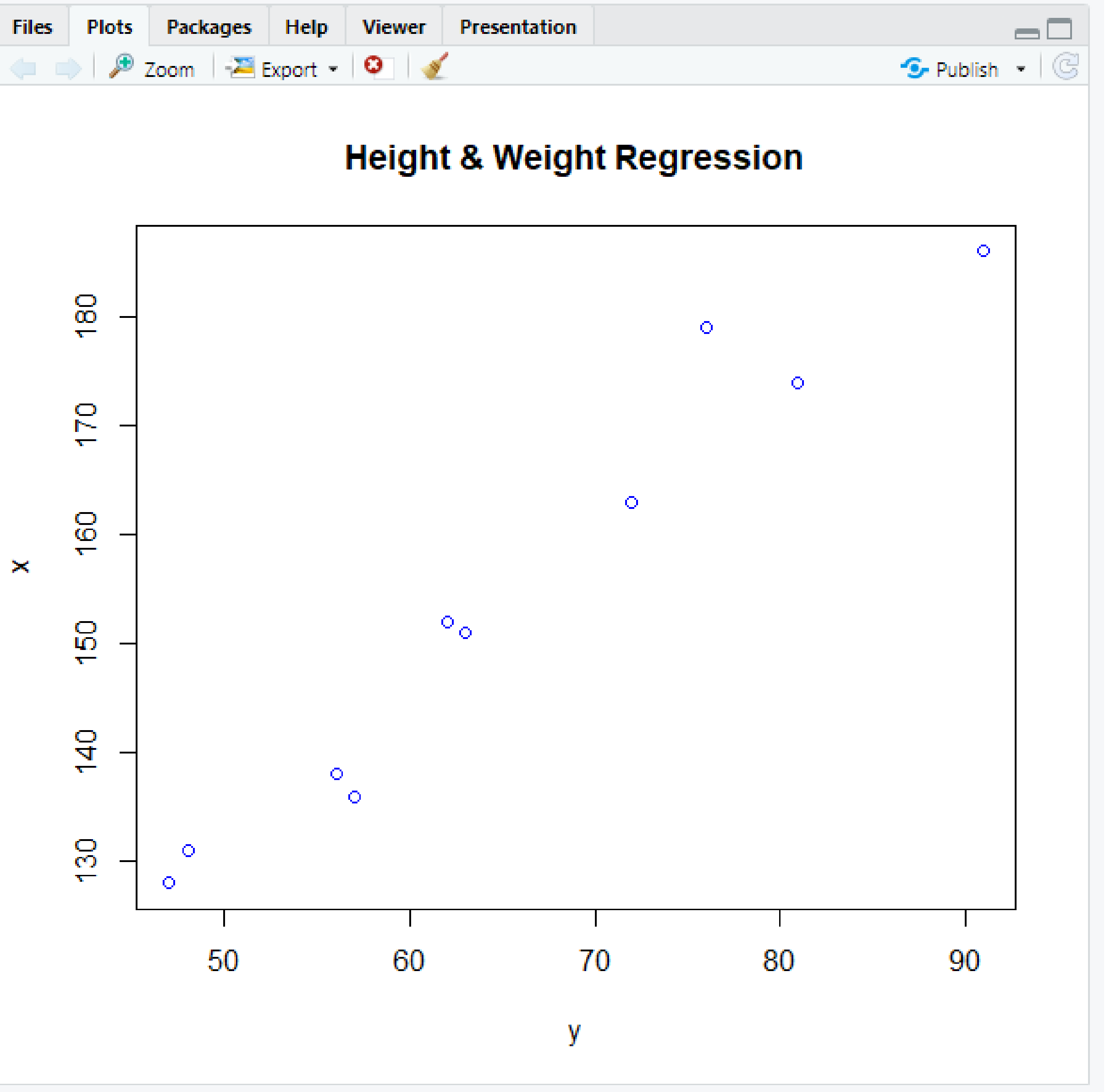
```
> a<-data.frame(x=170)
> result<-predict(relation,a)
> print(result)
       1
76.22869
```

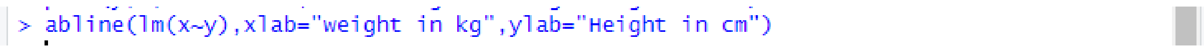
d. Visualize the regression graphically.



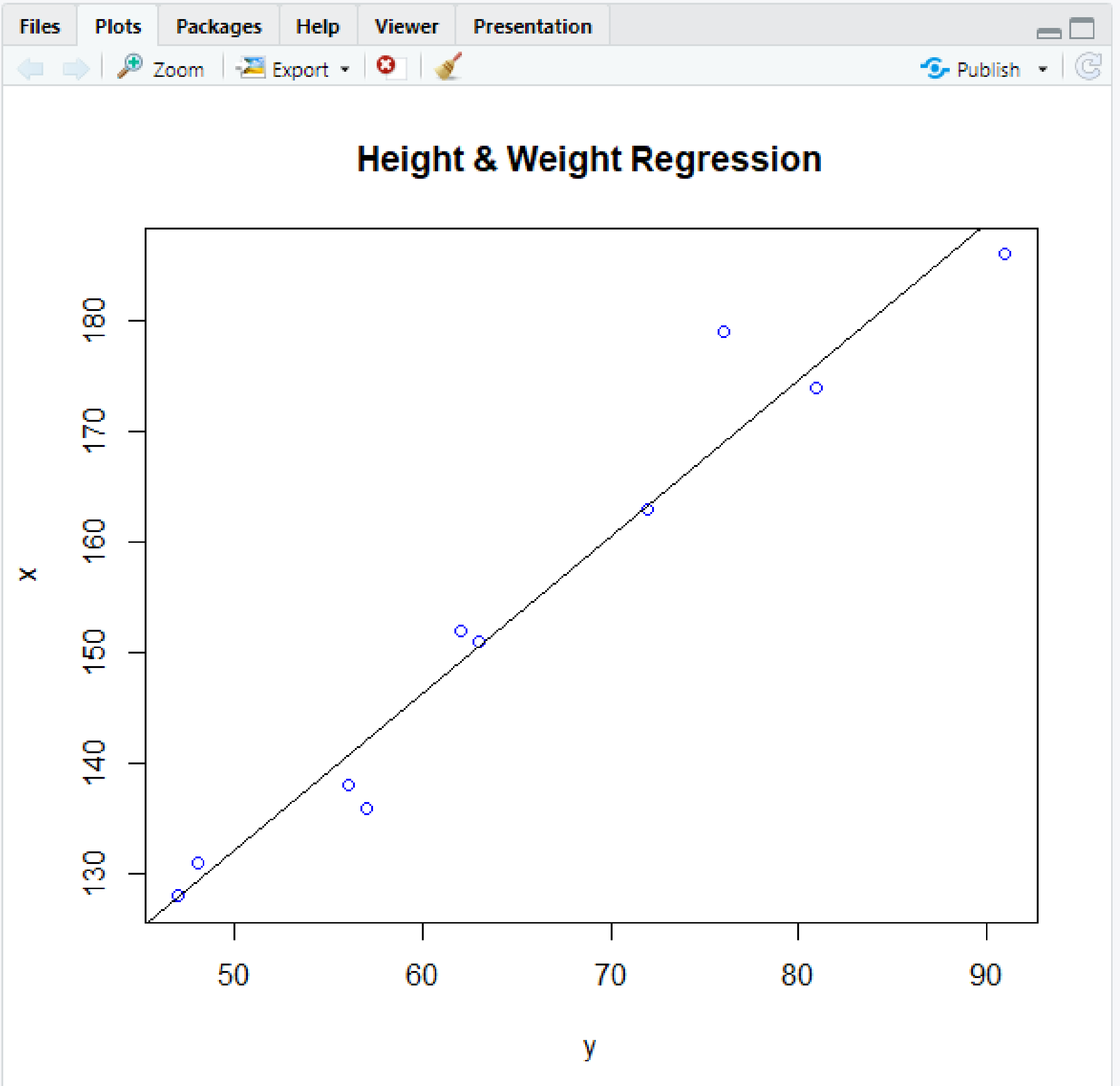
```
> plot(y,x,col="blue",main="Height & Weight Regression")
```

AkashkumarPrasad Rollno:80





```
> abline(lm(x~y),xlab="weight in kg",ylab="Height in cm")
```

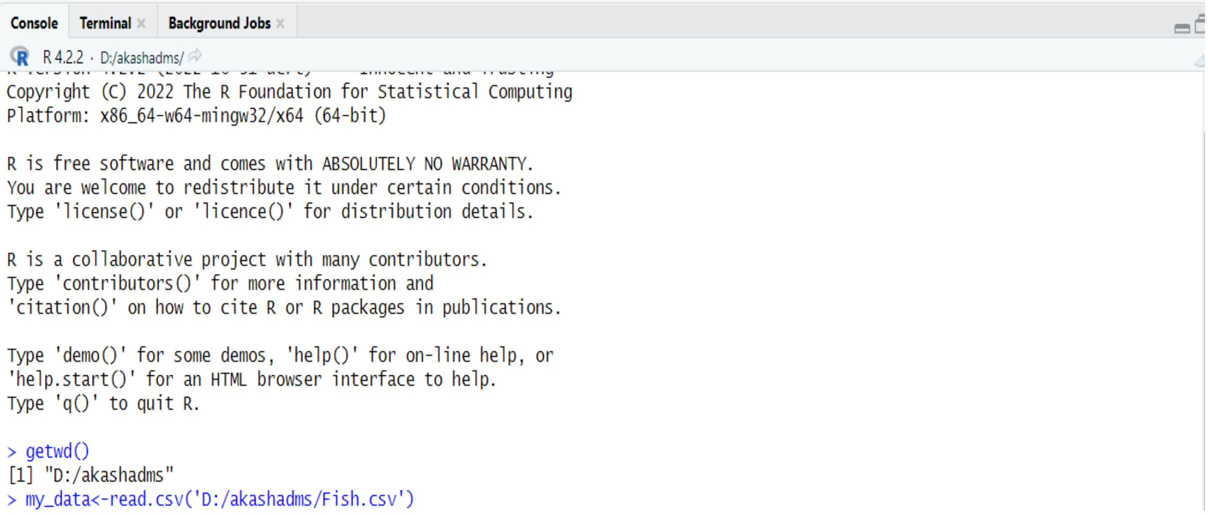


2. Simple Linear regression

Follow below step to implement Simple Linear regression on given database

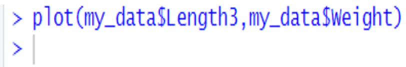
a. Use the dataset Fish.csv for linear regression

AkashkumarPrasad Rollno:80

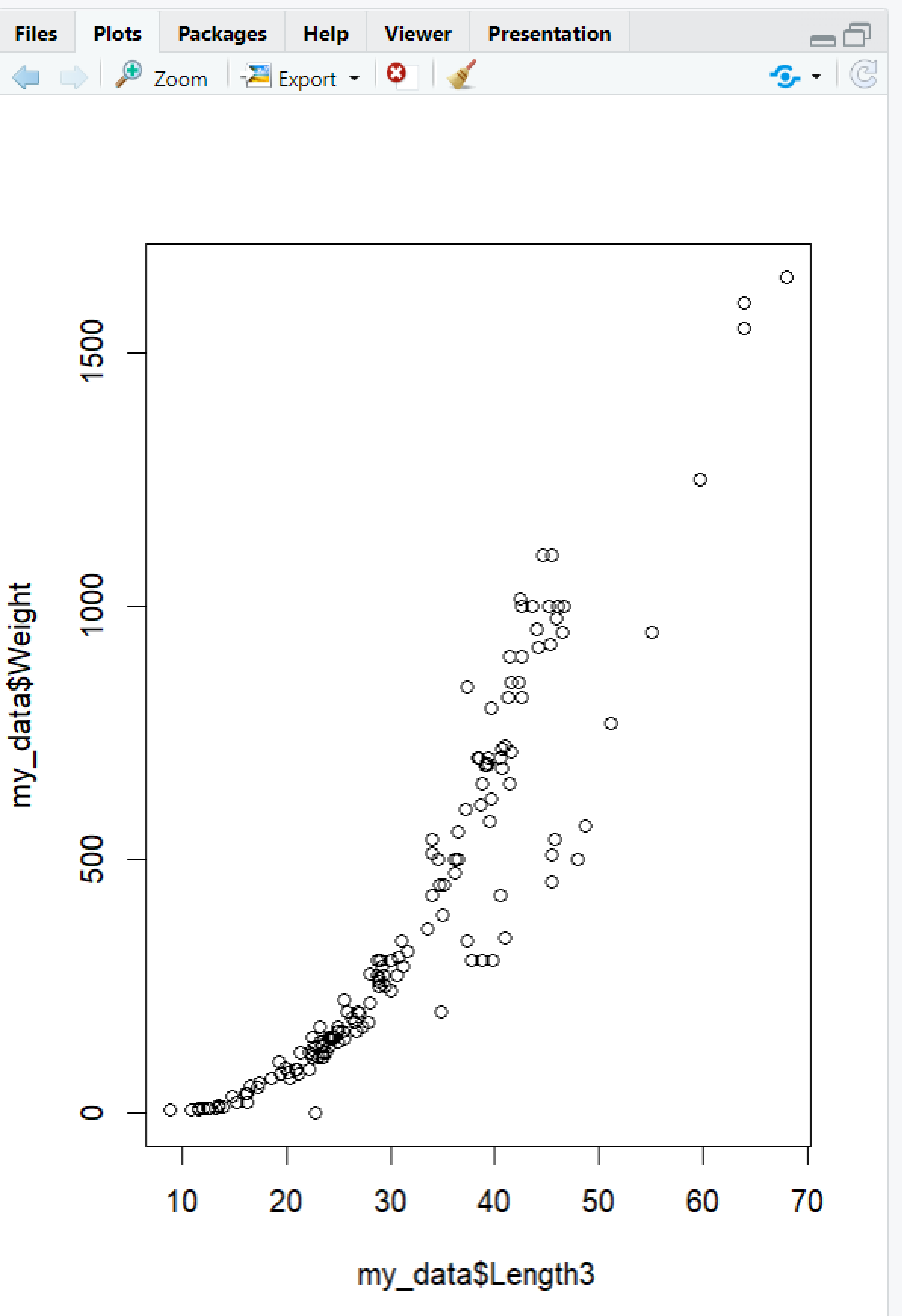


```
Copyright (C) 2022 The R Foundation for Statistical Computing
Platform: x86_64-w64-mingw32/x64 (64-bit)

R is free software and comes with ABSOLUTELY NO WARRANTY.
You are welcome to redistribute it under certain conditions.
Type 'license()' or 'licence()' for distribution details.

R is a collaborative project with many contributors.
Type 'contributors()' for more information and
'citation()' on how to cite R or R packages in publications.

Type 'demo()' for some demos, 'help()' for on-line help, or
'help.start()' for an HTML browser interface to help.
Type 'q()' to quit R.

> getwd()
[1] "D:/akashadms"
> my_data<-read.csv('D:/akashadms/Fish.csv')
```

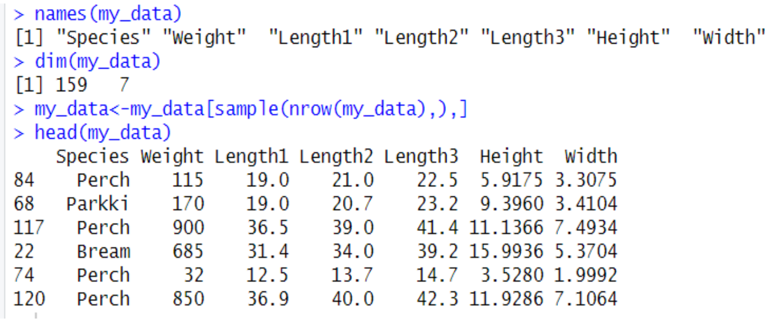
Plot the scatter graphs and check the relationship between Length3 and Weight columns
of Fish dataset



```
> plot(my_data$Length3,my_data$Weight)
> 
```

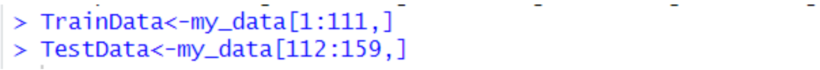


Randomize the dataset rows



```
> names(my_data)
[1] "Species" "Weight"  "Length1" "Length2" "Length3" "Height"  "Width"
> dim(my_data)
[1] 159   7
> my_data<-my_data[sample(nrow(my_data),),]
> head(my_data)
    Species Weight Length1 Length2 Length3  Height  Width
84    Perch    115    19.0    21.0    22.5  5.9175 3.3075
68   Parkki    170    19.0    20.7    23.2  9.3960 3.4104
117   Perch    900    36.5    39.0    41.4 11.1366 7.4934
22    Bream    685    31.4    34.0    39.2 15.9936 5.3704
74    Perch     32    12.5    13.7    14.7  3.5280 1.9992
120   Perch    850    36.9    40.0    42.3 11.9286 7.1064
```

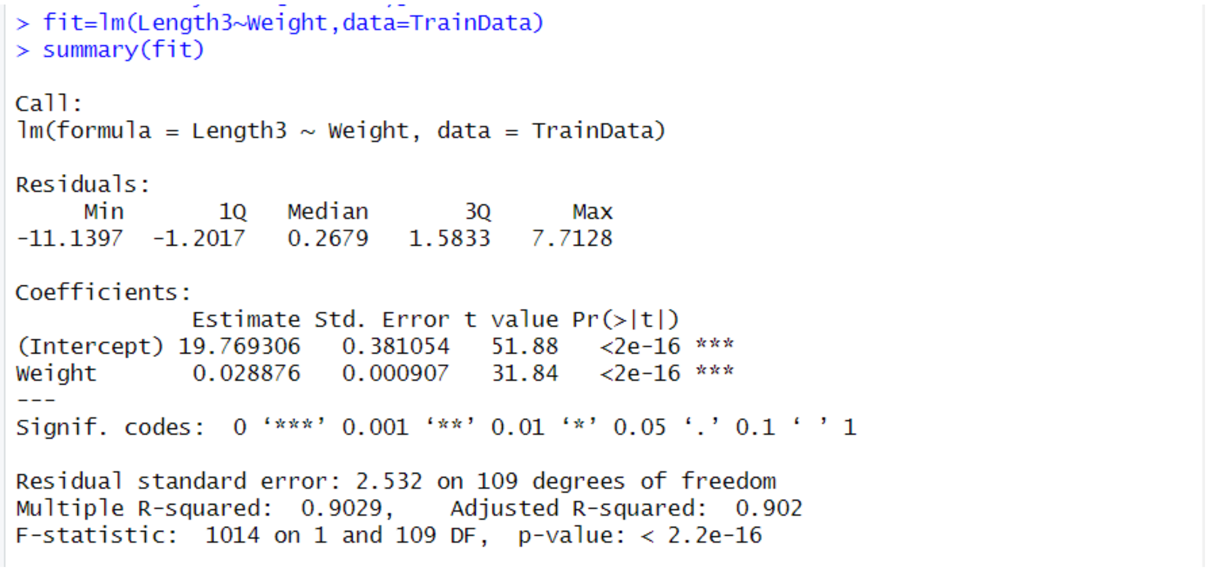
Split the data set into Training Data set and Test Data set.



```
> TrainData<-my_data[1:111,]
> TestData<-my_data[112:159,]
```

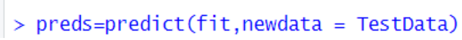
e. Perform single linear regression analysis on training dataset columns Length3 as Y and

Weight as X, using linear model function (lm).



```
> fit=lm(Length3~Weight,data=TrainData)
> summary(fit)

Call:
lm(formula = Length3 ~ Weight, data = TrainData)

Residuals:
     Min       1Q   Median       3Q      Max 
-11.1397  -1.2017   0.2679   1.5833   7.7128 

Coefficients:
             Estimate Std. Error t value Pr(>|t|)    
(Intercept) 19.769306   0.381054   51.88   <2e-16 ***
Weight       0.028876   0.000907   31.84   <2e-16 ***
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Residual standard error: 2.532 on 109 degrees of freedom
Multiple R-squared:  0.9029,	Adjusted R-squared:  0.902 
F-statistic:  1014 on 1 and 109 DF,  p-value: < 2.2e-16
```

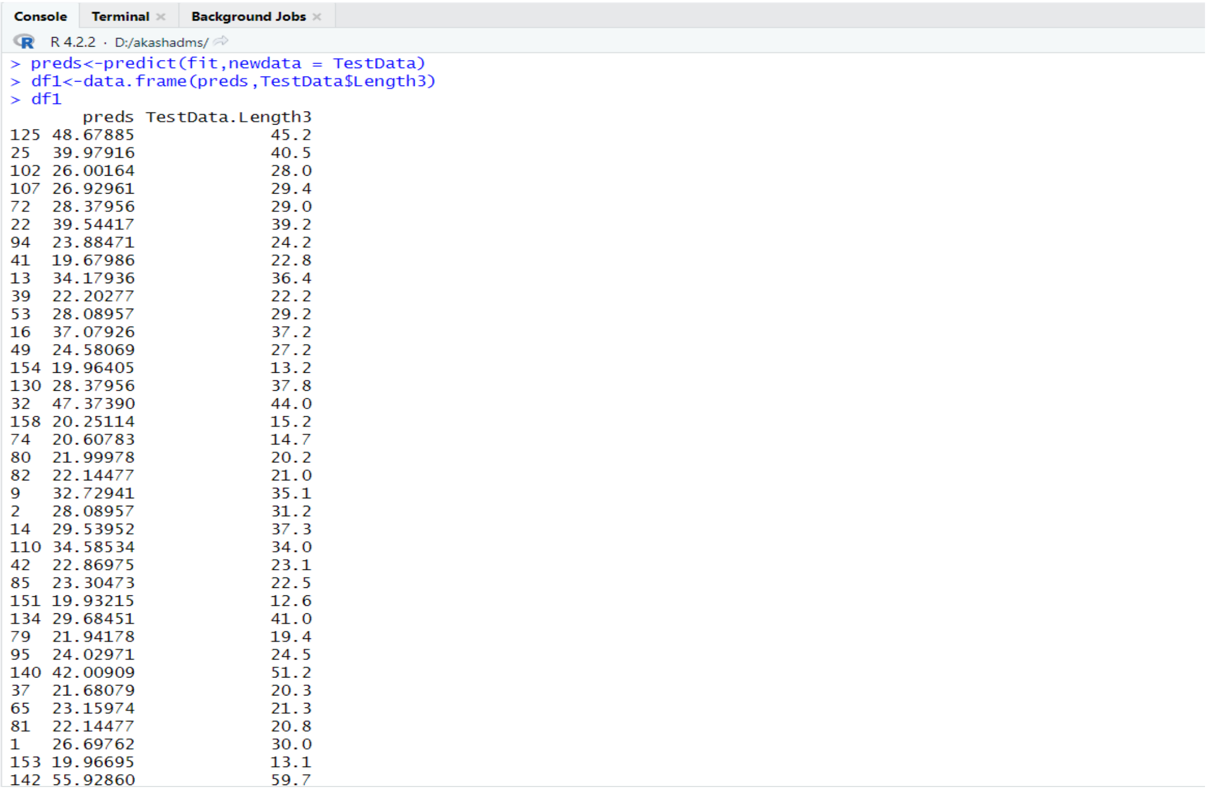
f. Predict the Length3 value using Testing dataset



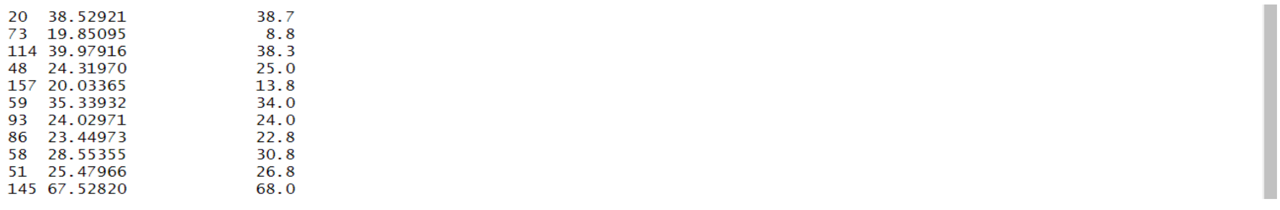
```
> preds=predict(fit,newdata = TestData)
```

g.Analyze the Testing result using predicted and actual value of the Length3 column data and calculate correlation between them

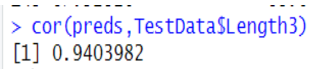
AkashkumarPrasad Rollno:80



```
> preds<-predict(fit,newdata = TestData)
> df1<-data.frame(preds,TestData$Length3)
> df1
       preds TestData.Length3
125 48.67885             45.2
25  39.97916             40.5
102 26.00164             28.0
107 26.92961             29.4
72  28.37956             29.0
22  39.54417             39.2
94  23.88471             24.2
41  19.67986             22.8
13  34.17936             36.4
39  22.20277             22.2
53  28.08957             29.2
16  37.07926             37.2
49  24.58069             27.2
154 19.96405             13.2
130 28.37956             37.8
32  47.37390             44.0
158 20.25114             15.2
74  20.60783             14.7
80  21.99978             20.2
82  22.14477             21.0
9   32.72941             35.1
2   28.08957             31.2
14  29.53952             37.3
110 34.58534             34.0
42  22.86975             23.1
85  23.30473             22.5
151 19.93215             12.6
134 29.68451             41.0
79  21.94178             19.4
95  24.02971             24.5
140 42.00909             51.2
37  21.68079             20.3
65  23.15974             21.3
81  22.14477             20.8
1   26.69762             30.0
153 19.96695             13.1
142 55.92860             59.7
```



```
20  38.52921             38.7
73  19.85095              8.8
114 39.97916             38.3
48  24.31970             25.0
157 20.03365             13.8
59  35.33932             34.0
93  24.02971             24.0
86  23.44973             22.8
58  28.55355             30.8
51  25.47966             26.8
145 67.52820             68.0
```

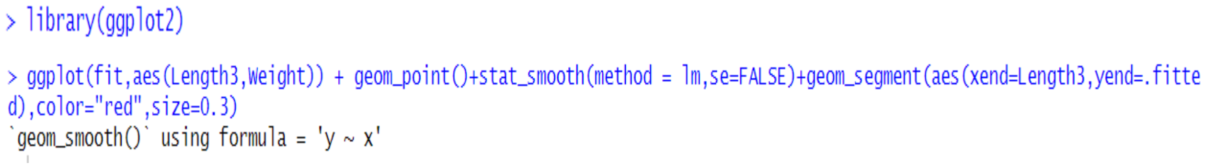


```
> cor(preds,TestData$Length3)
[1] 0.9403982
```

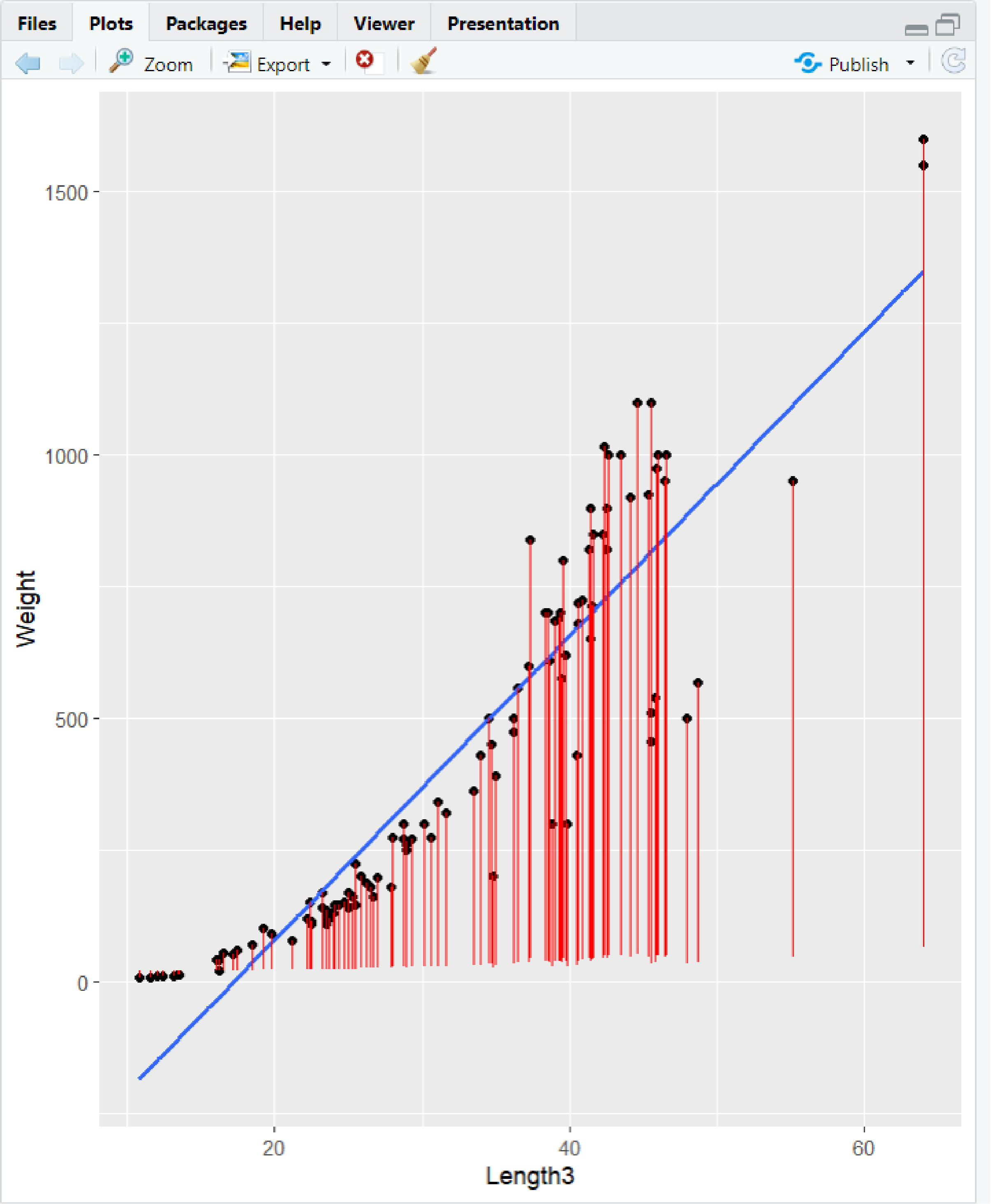
h.Analyze the regression line with Residuals(line segment which represents the distance

between y-value of the actual scatter plot points and the y values of the regression

equation at those points) on a scatter plot



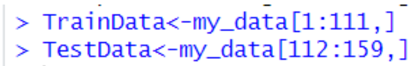
```
> library(ggplot2)
> ggplot(fit,aes(Length3,Weight)) + geom_point()+stat_smooth(method = lm,se=FALSE)+geom_segment(aes(xend=Length3,yend=.fitted),color="red",size=0.3)
`geom_smooth()` using formula = 'y ~ x'
```



Multiple Linear regression

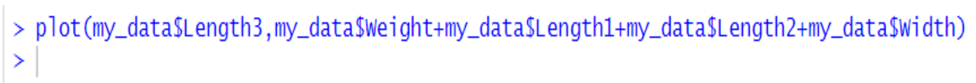
Follow below step to implement Multiple Linear regression on given database

a. Use the same training and testing dataset of Fish.csv created in exercise 2.

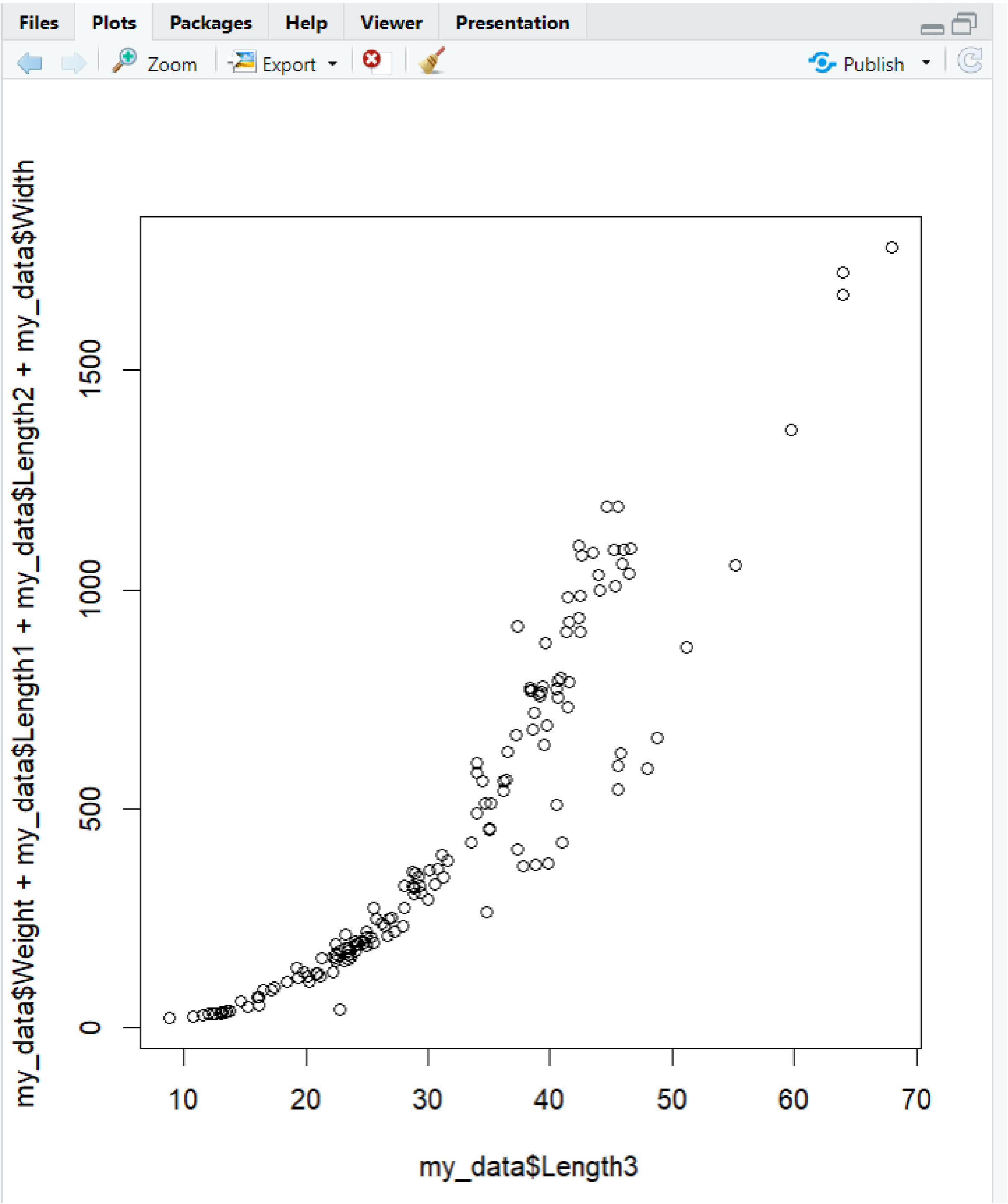


```
> TrainData<-my_data[1:111,]
> TestData<-my_data[112:159,]
```

b. Plot the scatter graphs and check the relationship between (Length3) and (Weight,
Length1, Length2, Width) columns



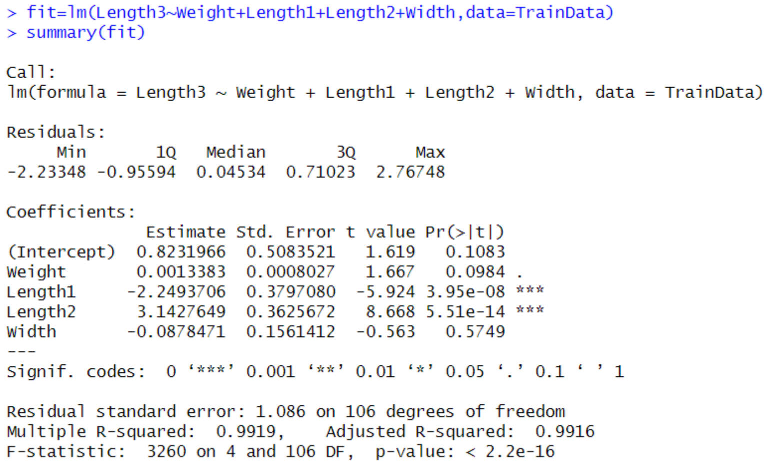
```
> plot(my_data$Length3,my_data$Weight+my_data$Length1+my_data$Length2+my_data$Width)
> 
```



c.Perform multiple regression analysis on training dataset columns Length3 as Y and

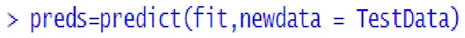
Weight, Length2, Length1, Width as X1, X2, X3, X4, using linear model function (lm).

AkashkumarPrasad Rollno:80



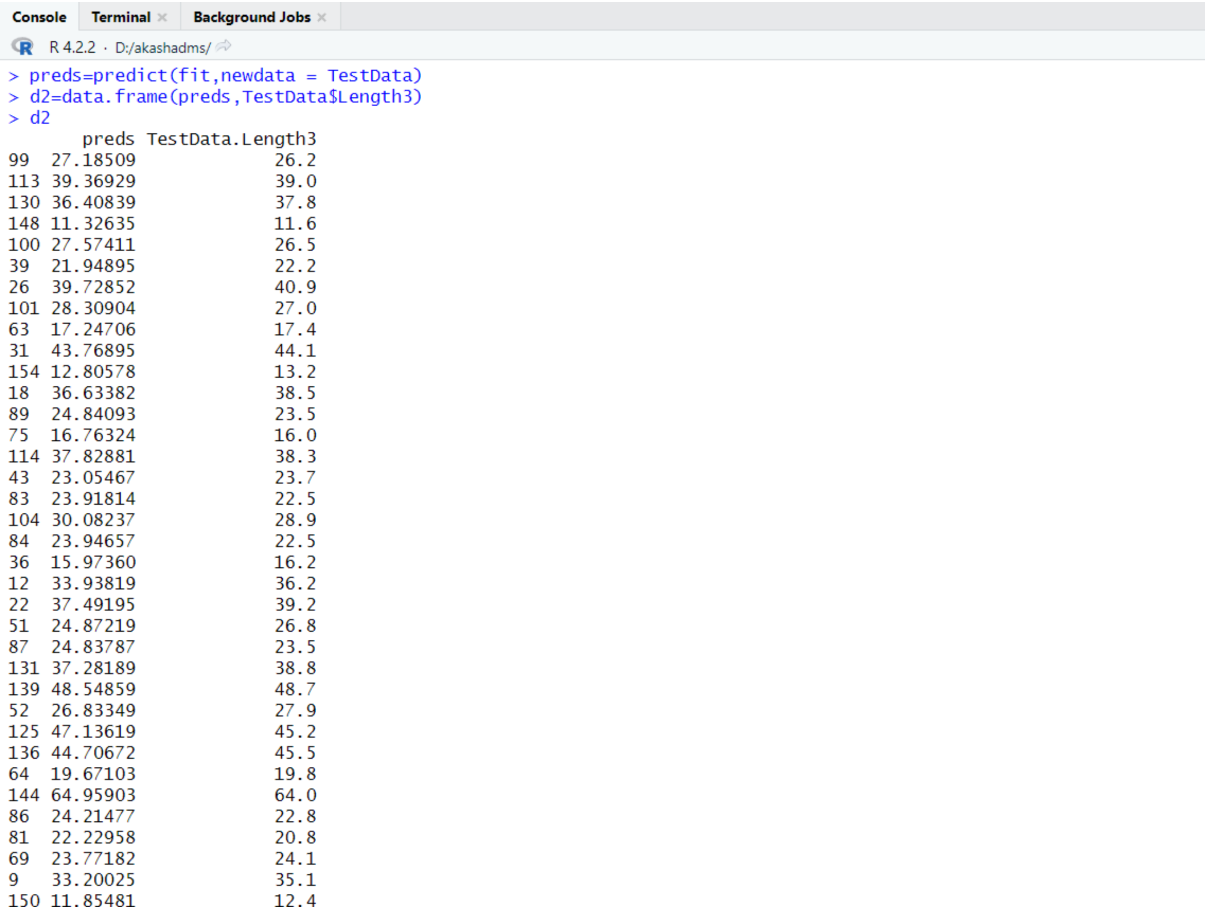
```
> fit=lm(Length3~Weight+Length1+Length2+Width,data=TrainData)
> summary(fit)

Call:
lm(formula = Length3 ~ Weight + Length1 + Length2 + Width, data = TrainData)

Residuals:
     Min       1Q   Median       3Q      Max 
-2.23348 -0.95594  0.04534  0.71023  2.76748 

Coefficients:
              Estimate Std. Error t value Pr(>|t|)    
(Intercept)  0.8231966  0.5083521   1.619   0.1083    
Weight       0.0013383  0.0008027   1.667   0.0984 .  
Length1     -2.2493706  0.3797080  -5.924 3.95e-08 ***
Length2      3.1427649  0.3625672   8.668 5.51e-14 ***
Width       -0.0878471  0.1561412  -0.563   0.5749    
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Residual standard error: 1.086 on 106 degrees of freedom
Multiple R-squared:  0.9919,	Adjusted R-squared:  0.9916 
F-statistic:  3260 on 4 and 106 DF,  p-value: < 2.2e-16
```

d.Predict the Length3 value using Testing dataset

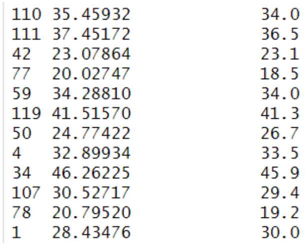


```
> preds=predict(fit,newdata = TestData)
```

e.Analyze the Testing result using predicted and actual value of the Length3 column data
and calculate correlation between them



```
> preds=predict(fit,newdata = TestData)
> d2=data.frame(preds,TestData$Length3)
> d2
       preds TestData.Length3
99  27.18509             26.2
113 39.36929             39.0
130 36.40839             37.8
148 11.32635             11.6
100 27.57411             26.5
39  21.94895             22.2
26  39.72852             40.9
101 28.30904             27.0
63  17.24706             17.4
31  43.76895             44.1
154 12.80578             13.2
18  36.63382             38.5
89  24.84093             23.5
75  16.76324             16.0
114 37.82881             38.3
43  23.05467             23.7
83  23.91814             22.5
104 30.08237             28.9
84  23.94657             22.5
36  15.97360             16.2
12  33.93819             36.2
22  37.49195             39.2
51  24.87219             26.8
87  24.83787             23.5
131 37.28189             38.8
139 48.54859             48.7
52  26.83349             27.9
125 47.13619             45.2
136 44.70672             45.5
64  19.67103             19.8
144 64.95903             64.0
86  24.21477             22.8
81  22.22958             20.8
69  23.77182             24.1
9   33.20025             35.1
150 11.85481             12.4
```

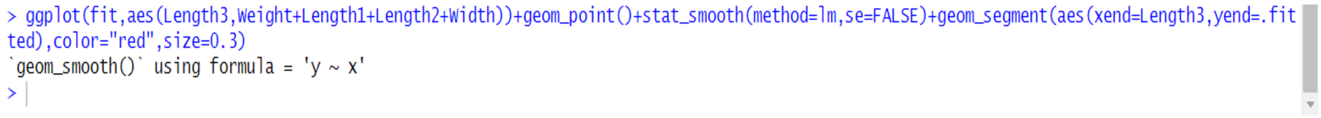


```
110 35.45932            34.0
111 37.45172            36.5
42  23.07864            23.1
77  20.02747            18.5
59  34.28810            34.0
119 41.51570            41.3
50  24.77422            26.7
4   32.89934            33.5
34  46.26225            45.9
107 30.52717            29.4
78  20.79520            19.2
1   28.43476            30.0
```

f.Analyze the regression line with Residuals(line segment which represents the distance

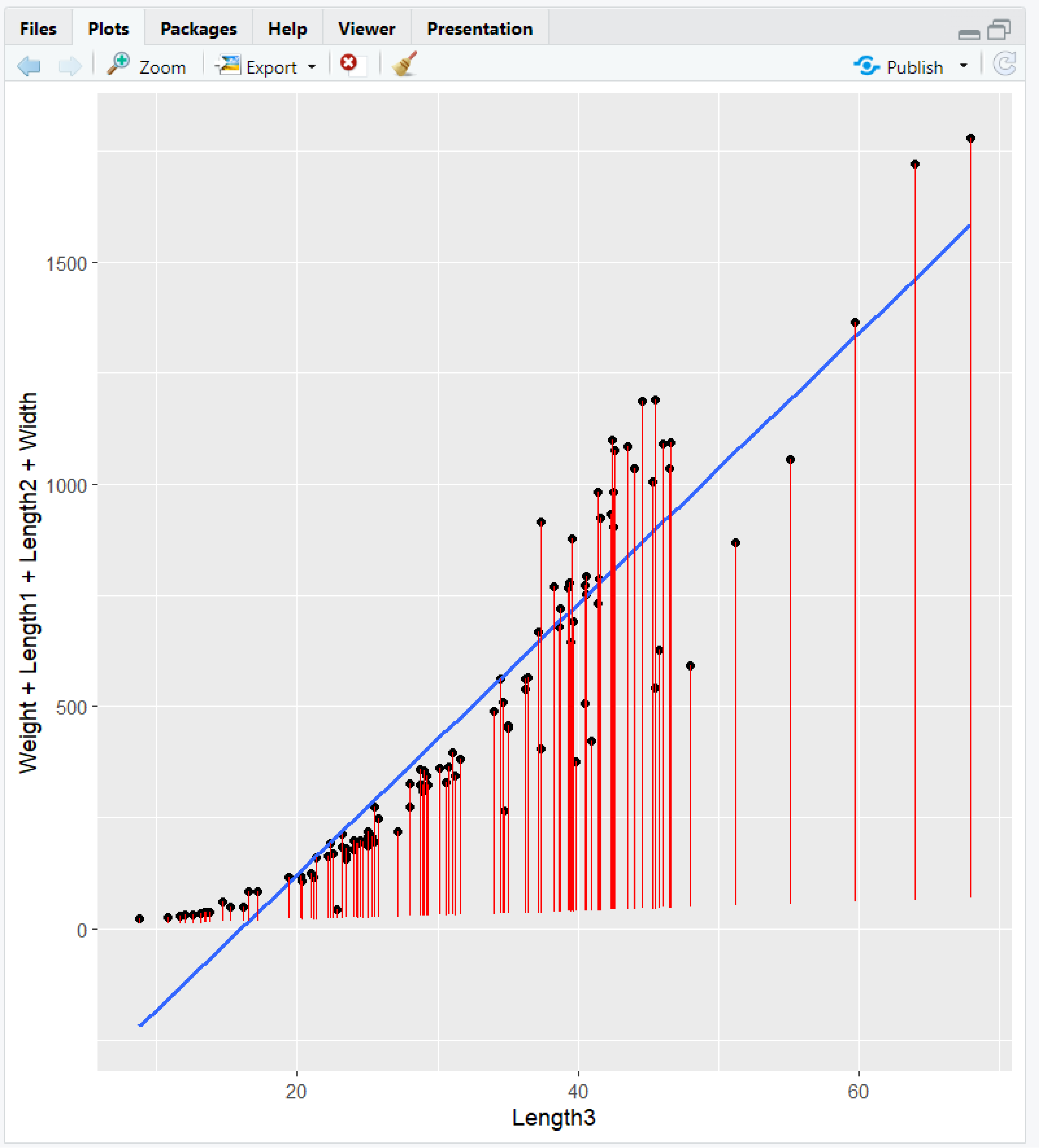
between y-value of the actual scatter plot points and the y values of the regression

equation at those points) on a scatter plot



```
> ggplot(fit,aes(Length3,Weight+Length1+Length2+Width))+geom_point()+stat_smooth(method=lm,se=FALSE)+geom_segment(aes(xend=Length3,yend=.fit
ted),color="red",size=0.3)
`geom_smooth()` using formula = 'y ~ x'
> 
```

AkashkumarPrasad Rollno:80


Files
Plots
Packages
Help
Viewer
Presentation
Zoom
Export
Publish
Weight + Length1 + Length2 + Width
1500
1000
500
0
20
40
60
Length3