predict weight gained using calories consumed

library(readr)  
data<-read.csv("E:\\assignments\\3.simple linear regression\\calories\_consumed.csv")  
data

## Weight.gained..grams. Calories.Consumed  
## 1 108 1500  
## 2 200 2300  
## 3 900 3400  
## 4 200 2200  
## 5 300 2500  
## 6 110 1600  
## 7 128 1400  
## 8 62 1900  
## 9 600 2800  
## 10 1100 3900  
## 11 100 1670  
## 12 150 1900  
## 13 350 2700  
## 14 700 3000

attach(data)  
WG<-Weight.gained..grams.  
CCON<-Calories.Consumed  
  
# Exploratory data analysis  
# structure of data  
str(data)

## 'data.frame': 14 obs. of 2 variables:  
## $ Weight.gained..grams.: int 108 200 900 200 300 110 128 62 600 1100 ...  
## $ Calories.Consumed : int 1500 2300 3400 2200 2500 1600 1400 1900 2800 3900 ...

# Descriptive statistics  
library(psych)  
describe(data)

## vars n mean sd median trimmed mad min max  
## Weight.gained..grams. 1 14 357.71 333.69 200 320.50 148.26 62 1100  
## Calories.Consumed 2 14 2340.71 752.11 2250 2289.17 837.67 1400 3900  
## range skew kurtosis se  
## Weight.gained..grams. 1038 1.00 -0.51 89.18  
## Calories.Consumed 2500 0.52 -0.93 201.01

class(data)

## [1] "data.frame"

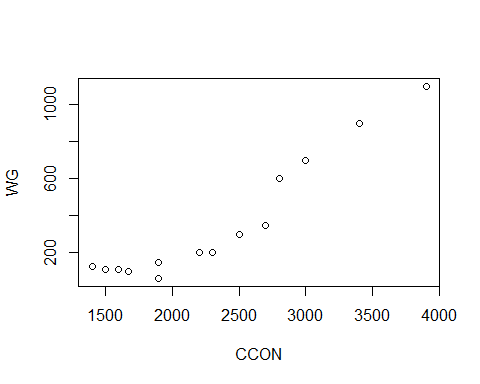
dim(data)

## [1] 14 2

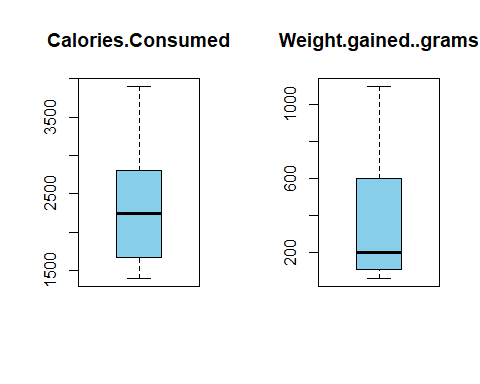
#correlation  
cor(CCON, WG)

## [1] 0.946991

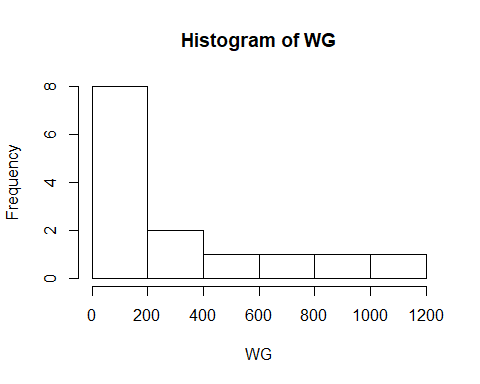
#scatter diagream  
plot(CCON, WG)



par(mfrow=c(1,2))  
boxplot(CCON, main="Calories.Consumed", col="skyblue")  
boxplot(WG, main="Weight.gained..grams", col="skyblue")



# from above boxplots there is no outliers  
hist(WG)



# the data is right skewed  
  
#Regression analysis  
reg<-lm(WG ~CCON )  
summary(reg)

##   
## Call:  
## lm(formula = WG ~ CCON)  
##   
## Residuals:  
## Min 1Q Median 3Q Max   
## -158.67 -107.56 36.70 81.68 165.53   
##   
## Coefficients:  
## Estimate Std. Error t value Pr(>|t|)   
## (Intercept) -625.75236 100.82293 -6.206 4.54e-05 \*\*\*  
## CCON 0.42016 0.04115 10.211 2.86e-07 \*\*\*  
## ---  
## Signif. codes: 0 '\*\*\*' 0.001 '\*\*' 0.01 '\*' 0.05 '.' 0.1 ' ' 1  
##   
## Residual standard error: 111.6 on 12 degrees of freedom  
## Multiple R-squared: 0.8968, Adjusted R-squared: 0.8882   
## F-statistic: 104.3 on 1 and 12 DF, p-value: 2.856e-07

pred\_val<-predict(reg)  
pred\_val

## 1 2 3 4 5 6   
## 4.482599 340.607908 802.780209 298.592245 424.639236 46.498263   
## 7 8 9 10 11 12   
## -37.533065 172.545254 550.686227 1012.858527 75.909227 172.545254   
## 13 14   
## 508.670563 634.717554

rmse<-sqrt(mean((pred\_val-WG)^2))  
rmse

## [1] 103.3025

confint(reg, level=0.95)

## 2.5 % 97.5 %  
## (Intercept) -845.4266546 -406.0780569  
## CCON 0.3305064 0.5098069

predict(reg, interval="predict")

## Warning in predict.lm(reg, interval = "predict"): predictions on current data refer to \_future\_ responses

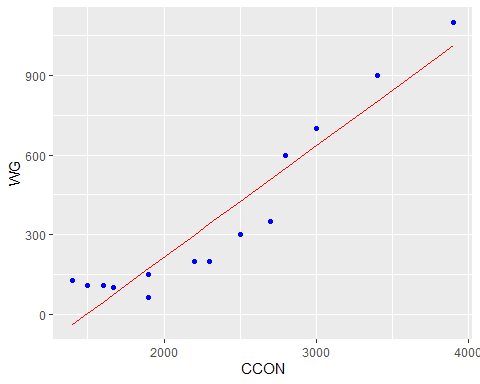
## fit lwr upr  
## 1 4.482599 -258.20569 267.1709  
## 2 340.607908 88.93791 592.2779  
## 3 802.780209 533.81393 1071.7465  
## 4 298.592245 46.63271 550.5518  
## 5 424.639236 172.59086 676.6876  
## 6 46.498263 -213.75953 306.7561  
## 7 -37.533065 -302.93258 227.8664  
## 8 172.545254 -82.18110 427.2716  
## 9 550.686227 295.69632 805.6761  
## 10 1012.858527 724.99432 1300.7227  
## 11 75.909227 -182.81852 334.6370  
## 12 172.545254 -82.18110 427.2716  
## 13 508.670563 254.97398 762.3671  
## 14 634.717554 376.22600 893.2091

library(ggplot2)

##   
## Attaching package: 'ggplot2'

## The following objects are masked from 'package:psych':  
##   
## %+%, alpha

ggplot(data=data, aes(x=CCON, y=WG))+geom\_point(color="blue")+geom\_line(color="red", data=data, aes(x=CCON, y=pred\_val))



# p vlaue<0.05 and multiple R-squared value is 0.8968  
# for multiple R-squared value uding trasformations  
#logarthmic Transformation  
reg\_log<-lm(WG ~ log(CCON))  
summary(reg\_log)

##   
## Call:  
## lm(formula = WG ~ log(CCON))  
##   
## Residuals:  
## Min 1Q Median 3Q Max   
## -187.44 -142.96 23.13 113.20 213.82   
##   
## Coefficients:  
## Estimate Std. Error t value Pr(>|t|)   
## (Intercept) -6955.7 1030.9 -6.747 2.05e-05 \*\*\*  
## log(CCON) 948.4 133.6 7.100 1.25e-05 \*\*\*  
## ---  
## Signif. codes: 0 '\*\*\*' 0.001 '\*\*' 0.01 '\*' 0.05 '.' 0.1 ' ' 1  
##   
## Residual standard error: 152.3 on 12 degrees of freedom  
## Multiple R-squared: 0.8077, Adjusted R-squared: 0.7917   
## F-statistic: 50.4 on 1 and 12 DF, p-value: 1.248e-05

pred\_val1<-predict(reg\_log)  
pred\_val1

## 1 2 3 4 5 6 7 8   
## -19.99870 385.37711 756.06367 343.22032 464.45388 41.20781 -85.42959 204.18573   
## 9 10 11 12 13 14   
## 571.93160 886.18133 81.81708 204.18573 537.44155 637.36248

confint(reg\_log, level=0.95)

## 2.5 % 97.5 %  
## (Intercept) -9201.8063 -4709.494  
## log(CCON) 657.3251 1239.418

predict(reg\_log, interval="predict")

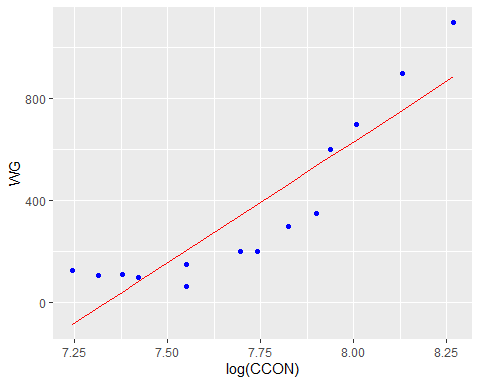
## Warning in predict.lm(reg\_log, interval = "predict"): predictions on current data refer to \_future\_ responses

## fit lwr upr  
## 1 -19.99870 -382.5178898 342.5205  
## 2 385.37711 41.7849717 728.9693  
## 3 756.06367 391.4700627 1120.6573  
## 4 343.22032 -0.2957275 686.7364  
## 5 464.45388 119.4081720 809.4996  
## 6 41.20781 -315.7491910 398.1648  
## 7 -85.42959 -454.8597180 284.0005  
## 8 204.18573 -142.5179686 550.8894  
## 9 571.93160 222.2096884 921.6535  
## 10 886.18133 506.3308457 1266.0318  
## 11 81.81708 -271.9519877 435.5861  
## 12 204.18573 -142.5179686 550.8894  
## 13 537.44155 189.5540023 885.3291  
## 14 637.36248 283.3161385 991.4088

rmse1<-sqrt(mean(reg\_log$residuals^2))  
rmse1

## [1] 141.0054

ggplot(data=data, aes(x=log(CCON), y=WG))+geom\_point(color="blue")+geom\_line(color="red", data=data, aes(x=log(CCON), y=pred\_val1))



# p value <0.05 and multiple R\_squared value is 0.8077  
# Exponential Transformation  
reg\_exp<-lm(log(WG) ~ CCON)  
summary(reg\_exp)

##   
## Call:  
## lm(formula = log(WG) ~ CCON)  
##   
## Residuals:  
## Min 1Q Median 3Q Max   
## -0.86537 -0.10532 0.02462 0.13467 0.42632   
##   
## Coefficients:  
## Estimate Std. Error t value Pr(>|t|)   
## (Intercept) 2.8386724 0.2994581 9.479 6.36e-07 \*\*\*  
## CCON 0.0011336 0.0001222 9.276 8.02e-07 \*\*\*  
## ---  
## Signif. codes: 0 '\*\*\*' 0.001 '\*\*' 0.01 '\*' 0.05 '.' 0.1 ' ' 1  
##   
## Residual standard error: 0.3314 on 12 degrees of freedom  
## Multiple R-squared: 0.8776, Adjusted R-squared: 0.8674   
## F-statistic: 86.04 on 1 and 12 DF, p-value: 8.018e-07

pred\_val2<-predict(reg\_exp)  
pred\_val2

## 1 2 3 4 5 6 7 8   
## 4.539069 5.445947 6.692904 5.332587 5.672666 4.652428 4.425709 4.992508   
## 9 10 11 12 13 14   
## 6.012745 7.259702 4.731780 4.992508 5.899386 6.239465

confint(reg\_exp, level=0.95)

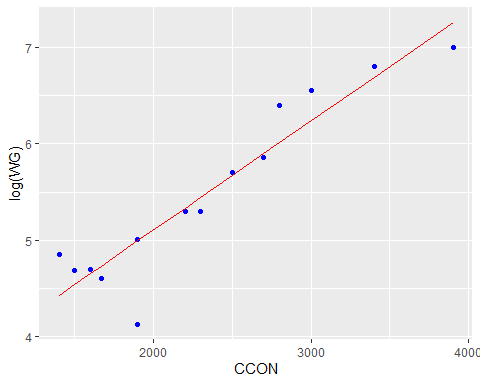
## 2.5 % 97.5 %  
## (Intercept) 2.1862091856 3.491135698  
## CCON 0.0008673238 0.001399871

predict(reg\_exp, interval="predict")

## Warning in predict.lm(reg\_exp, interval = "predict"): predictions on current data refer to \_future\_ responses

## fit lwr upr  
## 1 4.539069 3.758848 5.319289  
## 2 5.445947 4.698452 6.193442  
## 3 6.692904 5.894036 7.491771  
## 4 5.332587 4.584232 6.080942  
## 5 5.672666 4.924047 6.421285  
## 6 4.652428 3.879426 5.425430  
## 7 4.425709 3.637435 5.213982  
## 8 4.992508 4.235935 5.749080  
## 9 6.012745 5.255390 6.770101  
## 10 7.259702 6.404706 8.114699  
## 11 4.731780 3.963323 5.500238  
## 12 4.992508 4.235935 5.749080  
## 13 5.899386 5.145871 6.652900  
## 14 6.239465 5.471709 7.007221

ggplot(data=data, aes(x=CCON, y=log(WG)))+geom\_point(color="blue")+geom\_line(color="red", data=data, aes(x=CCON, y=pred\_val2))



#p value<0.05 and multiple R-squared value is 0.8776  
# polynomial of second degree transformation  
reg\_poly<-lm(WG ~ CCON+I(CCON^2))  
summary(reg\_poly)

##   
## Call:  
## lm(formula = WG ~ CCON + I(CCON^2))  
##   
## Residuals:  
## Min 1Q Median 3Q Max   
## -90.321 -63.843 -3.609 52.120 120.222   
##   
## Coefficients:  
## Estimate Std. Error t value Pr(>|t|)   
## (Intercept) 2.033e+02 2.436e+02 0.834 0.42185   
## CCON -2.919e-01 2.021e-01 -1.444 0.17653   
## I(CCON^2) 1.395e-04 3.918e-05 3.561 0.00447 \*\*  
## ---  
## Signif. codes: 0 '\*\*\*' 0.001 '\*\*' 0.01 '\*' 0.05 '.' 0.1 ' ' 1  
##   
## Residual standard error: 79.43 on 11 degrees of freedom  
## Multiple R-squared: 0.9521, Adjusted R-squared: 0.9433   
## F-statistic: 109.2 on 2 and 11 DF, p-value: 5.546e-08

pred\_val3<-predict(reg\_poly)  
pred\_val3

## 1 2 3 4 5 6 7   
## 79.33345 269.95364 823.64945 236.35984 345.51232 93.39470 68.06257   
## 8 9 10 11 12 13 14   
## 152.32063 479.77809 1186.94394 104.89784 152.32063 432.23247 583.24042

confint(reg\_poly, level=0.95)

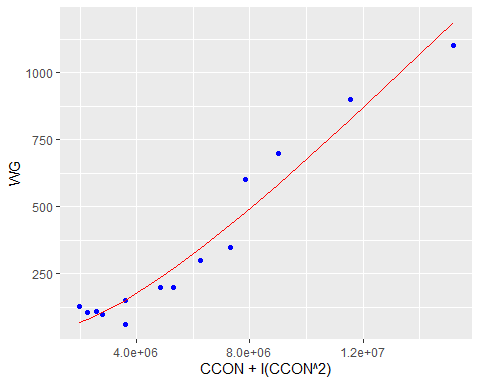
## 2.5 % 97.5 %  
## (Intercept) -3.329553e+02 7.394722e+02  
## CCON -7.367129e-01 1.529250e-01  
## I(CCON^2) 5.328126e-05 2.257552e-04

predict(reg\_poly, interval="predict")

## Warning in predict.lm(reg\_poly, interval = "predict"): predictions on current data refer to \_future\_ responses

## fit lwr upr  
## 1 79.33345 -115.15333 273.8202  
## 2 269.95364 83.77882 456.1285  
## 3 823.64945 629.80153 1017.4974  
## 4 236.35984 51.13324 421.5864  
## 5 345.51232 157.77722 533.2474  
## 6 93.39470 -95.99261 282.7820  
## 7 68.06257 -133.64283 269.7680  
## 8 152.32063 -31.28353 335.9248  
## 9 479.77809 291.24523 668.3109  
## 10 1186.94394 953.63989 1420.2480  
## 11 104.89784 -82.01852 291.8142  
## 12 152.32063 -31.28353 335.9248  
## 13 432.23247 243.77631 620.6886  
## 14 583.24042 394.65118 771.8297

ggplot(data=data, aes(x=CCON+I(CCON^2), y=WG))+geom\_point(color="blue")+geom\_line(color="red", data=data, aes(x=CCON+I(CCON^2), y=pred\_val3))



#multiple R-squared value is 0.9521  
#polynomial of three degree transformation  
reg\_poly1<-lm(WG ~ CCON+I(CCON^2)+I(CCON^3))  
summary(reg\_poly1)

##   
## Call:  
## lm(formula = WG ~ CCON + I(CCON^2) + I(CCON^3))  
##   
## Residuals:  
## Min 1Q Median 3Q Max   
## -106.119 -25.806 -4.866 18.874 78.571   
##   
## Coefficients:  
## Estimate Std. Error t value Pr(>|t|)   
## (Intercept) 2.663e+03 6.464e+02 4.120 0.00208 \*\*  
## CCON -3.451e+00 8.151e-01 -4.234 0.00173 \*\*  
## I(CCON^2) 1.409e-03 3.243e-04 4.346 0.00145 \*\*  
## I(CCON^3) -1.608e-07 4.093e-08 -3.928 0.00283 \*\*  
## ---  
## Signif. codes: 0 '\*\*\*' 0.001 '\*\*' 0.01 '\*' 0.05 '.' 0.1 ' ' 1  
##   
## Residual standard error: 52.24 on 10 degrees of freedom  
## Multiple R-squared: 0.9811, Adjusted R-squared: 0.9755   
## F-statistic: 173.5 on 3 and 10 DF, p-value: 6.395e-09

pred\_val4<-predict(reg\_poly1)  
pred\_val4

## 1 2 3 4 5 6 7   
## 115.52138 225.96582 904.28877 181.02235 332.89581 91.43641 153.32619   
## 8 9 10 11 12 13 14   
## 91.85395 521.42855 1105.44313 82.22366 91.85395 456.11882 654.62121

confint(reg\_poly1, level=0.95)

## 2.5 % 97.5 %  
## (Intercept) 1.223057e+03 4.103715e+03  
## CCON -5.267244e+00 -1.634872e+00  
## I(CCON^2) 6.868757e-04 2.132096e-03  
## I(CCON^3) -2.519688e-07 -6.958405e-08

predict(reg\_poly1, interval="predict")

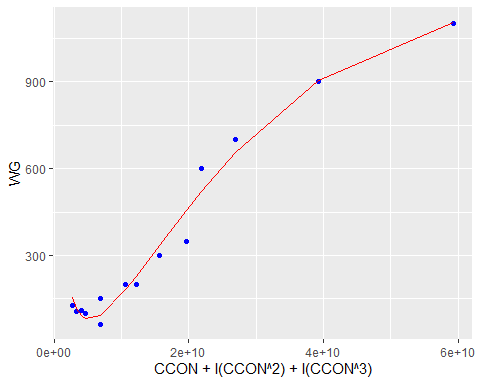
## Warning in predict.lm(reg\_poly1, interval = "predict"): predictions on current data refer to \_future\_ responses

## fit lwr upr  
## 1 115.52138 -15.58183 246.6246  
## 2 225.96582 99.52724 352.4044  
## 3 904.28877 767.36253 1041.2150  
## 4 181.02235 53.76953 308.2752  
## 5 332.89581 207.69983 458.0918  
## 6 91.43641 -34.65979 217.5326  
## 7 153.32619 10.59106 296.0613  
## 8 91.85395 -35.10714 218.8150  
## 9 521.42855 393.70236 649.1547  
## 10 1105.44313 943.37986 1267.5064  
## 11 82.22366 -42.88531 207.3326  
## 12 91.85395 -35.10714 218.8150  
## 13 456.11882 329.91810 582.3195  
## 14 654.62121 522.69503 786.5474

rmse<-sqrt(mean(reg\_poly1$residuals^2))  
rmse

## [1] 44.15011

ggplot(data=data, aes(x=CCON + I(CCON^2) + I(CCON^3), y=WG))+geom\_point(color="blue")+geom\_line(color="red", data=data, aes(x=CCON + I(CCON^2) + I(CCON^3), y=pred\_val4))



# p values<0.05, multiple R-squared value is 0.9811 and adjusted R-squared value is 0.9755  
#polynomial of Three degree gives the best R-squared values  
predicted\_wg\_val<-predict(reg\_poly1)  
predicted\_wg\_val

## 1 2 3 4 5 6 7   
## 115.52138 225.96582 904.28877 181.02235 332.89581 91.43641 153.32619   
## 8 9 10 11 12 13 14   
## 91.85395 521.42855 1105.44313 82.22366 91.85395 456.11882 654.62121

par(mfrow=c(2,2))  
plot(reg\_poly1)

