Predict delivery time using sorting time

library(readr)  
data<-read\_csv("E:\\assignments\\3.simple linear regression\\delivery\_time.csv")

## Parsed with column specification:  
## cols(  
## `Delivery Time` = col\_double(),  
## `Sorting Time` = col\_double()  
## )

View(data)  
attach(data)  
DT<-`Delivery Time`  
ST<-`Sorting Time`  
  
# Exploratory data analysis  
# structure of data  
head(data)

## # A tibble: 6 x 2  
## `Delivery Time` `Sorting Time`  
## <dbl> <dbl>  
## 1 21 10  
## 2 13.5 4  
## 3 19.8 6  
## 4 24 9  
## 5 29 10  
## 6 15.4 6

str(data)

## Classes 'spec\_tbl\_df', 'tbl\_df', 'tbl' and 'data.frame': 21 obs. of 2 variables:  
## $ Delivery Time: num 21 13.5 19.8 24 29 ...  
## $ Sorting Time : num 10 4 6 9 10 6 7 3 10 9 ...  
## - attr(\*, "spec")=  
## .. cols(  
## .. `Delivery Time` = col\_double(),  
## .. `Sorting Time` = col\_double()  
## .. )

dim(data)

## [1] 21 2

summary(data)

## Delivery Time Sorting Time   
## Min. : 8.00 Min. : 2.00   
## 1st Qu.:13.50 1st Qu.: 4.00   
## Median :17.83 Median : 6.00   
## Mean :16.79 Mean : 6.19   
## 3rd Qu.:19.75 3rd Qu.: 8.00   
## Max. :29.00 Max. :10.00

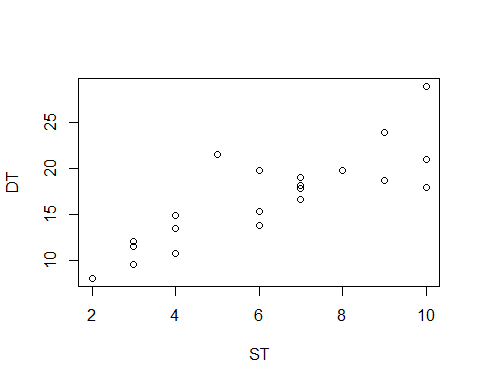
library(psych)  
describe(data)

## vars n mean sd median trimmed mad min max range skew  
## Delivery Time 1 21 16.79 5.07 17.83 16.59 4.70 8 29 21 0.30  
## Sorting Time 2 21 6.19 2.54 6.00 6.18 2.97 2 10 8 0.04  
## kurtosis se  
## Delivery Time -0.30 1.11  
## Sorting Time -1.34 0.55

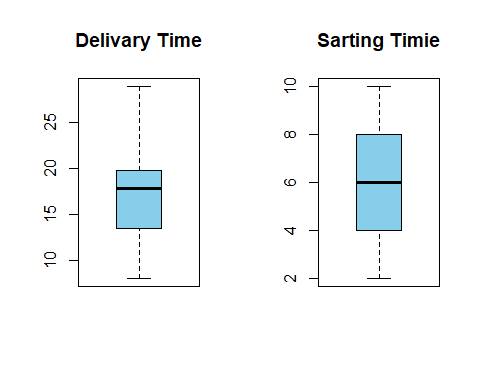
cor(ST,DT)

## [1] 0.8259973

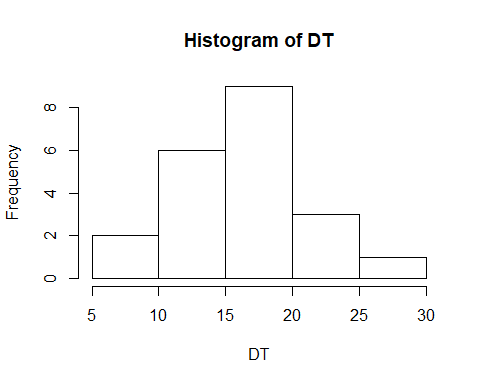
#scatter diagram  
plot(ST,DT)



#positively correlated  
par(mfrow=c(1,2))  
boxplot(DT, main="Delivary Time", col="skyblue")  
boxplot(ST, main="Sarting Timie", col="skyblue")



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



#right skewed  
# if |r|<0.85 , this is moderately correlated  
# Regression analysis  
reg<-lm(DT ~ ST)  
summary(reg)

##   
## Call:  
## lm(formula = DT ~ ST)  
##   
## Residuals:  
## Min 1Q Median 3Q Max   
## -5.1729 -2.0298 -0.0298 0.8741 6.6722   
##   
## Coefficients:  
## Estimate Std. Error t value Pr(>|t|)   
## (Intercept) 6.5827 1.7217 3.823 0.00115 \*\*   
## ST 1.6490 0.2582 6.387 3.98e-06 \*\*\*  
## ---  
## Signif. codes: 0 '\*\*\*' 0.001 '\*\*' 0.01 '\*' 0.05 '.' 0.1 ' ' 1  
##   
## Residual standard error: 2.935 on 19 degrees of freedom  
## Multiple R-squared: 0.6823, Adjusted R-squared: 0.6655   
## F-statistic: 40.8 on 1 and 19 DF, p-value: 3.983e-06

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

## 1 2 3 4 5 6 7 8   
## 23.072933 13.178814 16.476853 21.423913 23.072933 16.476853 18.125873 11.529794   
## 9 10 11 12 13 14 15 16   
## 23.072933 21.423913 19.774893 13.178814 18.125873 11.529794 11.529794 13.178814   
## 17 18 19 20 21   
## 16.476853 18.125873 9.880774 18.125873 14.827833

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

## [1] 2.79165

confint(reg, level=0.95)

## 2.5 % 97.5 %  
## (Intercept) 2.979134 10.186334  
## ST 1.108673 2.189367

predict(reg, interval="predict")

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

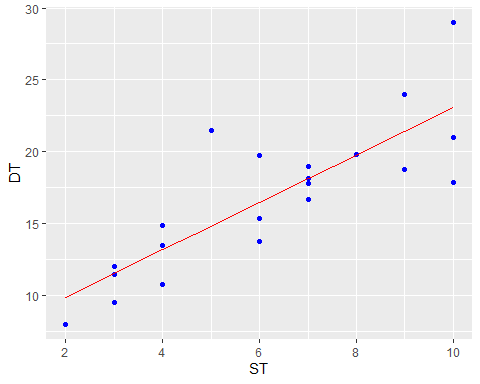
## fit lwr upr  
## 1 23.072933 16.457161 29.68870  
## 2 13.178814 6.780993 19.57663  
## 3 16.476853 10.188630 22.76508  
## 4 21.423913 14.955850 27.89198  
## 5 23.072933 16.457161 29.68870  
## 6 16.476853 10.188630 22.76508  
## 7 18.125873 11.823294 24.42845  
## 8 11.529794 5.010345 18.04924  
## 9 23.072933 16.457161 29.68870  
## 10 21.423913 14.955850 27.89198  
## 11 19.774893 13.411938 26.13785  
## 12 13.178814 6.780993 19.57663  
## 13 18.125873 11.823294 24.42845  
## 14 11.529794 5.010345 18.04924  
## 15 11.529794 5.010345 18.04924  
## 16 13.178814 6.780993 19.57663  
## 17 16.476853 10.188630 22.76508  
## 18 18.125873 11.823294 24.42845  
## 19 9.880774 3.198090 16.56346  
## 20 18.125873 11.823294 24.42845  
## 21 14.827833 8.507631 21.14804

library(ggplot2)

##   
## Attaching package: 'ggplot2'

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

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



# pval<0.05 and multiple R-square is 0.6823, for better multiple R-square value uing transformations  
#logarthmic Transformation  
reg\_log<-lm(DT ~ log(ST))  
summary(reg\_log)

##   
## Call:  
## lm(formula = DT ~ log(ST))  
##   
## Residuals:  
## Min 1Q Median 3Q Max   
## -4.0829 -2.0133 -0.1965 0.9351 7.0171   
##   
## Coefficients:  
## Estimate Std. Error t value Pr(>|t|)   
## (Intercept) 1.160 2.455 0.472 0.642   
## log(ST) 9.043 1.373 6.587 2.64e-06 \*\*\*  
## ---  
## Signif. codes: 0 '\*\*\*' 0.001 '\*\*' 0.01 '\*' 0.05 '.' 0.1 ' ' 1  
##   
## Residual standard error: 2.873 on 19 degrees of freedom  
## Multiple R-squared: 0.6954, Adjusted R-squared: 0.6794   
## F-statistic: 43.39 on 1 and 19 DF, p-value: 2.642e-06

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

## 1 2 3 4 5 6 7 8   
## 21.98291 13.69652 17.36331 21.03009 21.98291 17.36331 18.75735 11.09489   
## 9 10 11 12 13 14 15 16   
## 21.98291 21.03009 19.96493 13.69652 18.75735 11.09489 11.09489 13.69652   
## 17 18 19 20 21   
## 17.36331 18.75735 7.42810 18.75735 15.71450

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

## [1] 2.733171

confint(reg\_log, level=0.95)

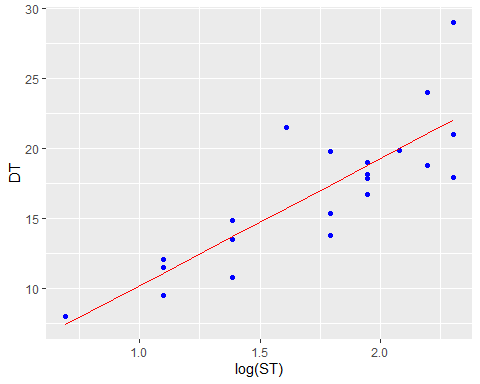
## 2.5 % 97.5 %  
## (Intercept) -3.97778 6.297147  
## log(ST) 6.16977 11.917057

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 21.98291 15.6099875 28.35584  
## 2 13.69652 7.4628028 19.93023  
## 3 17.36331 11.2049447 23.52167  
## 4 21.03009 14.7287585 27.33143  
## 5 21.98291 15.6099875 28.35584  
## 6 17.36331 11.2049447 23.52167  
## 7 18.75735 12.5700473 24.94466  
## 8 11.09489 4.6786298 17.51115  
## 9 21.98291 15.6099875 28.35584  
## 10 21.03009 14.7287585 27.33143  
## 11 19.96493 13.7271824 26.20268  
## 12 13.69652 7.4628028 19.93023  
## 13 18.75735 12.5700473 24.94466  
## 14 11.09489 4.6786298 17.51115  
## 15 11.09489 4.6786298 17.51115  
## 16 13.69652 7.4628028 19.93023  
## 17 17.36331 11.2049447 23.52167  
## 18 18.75735 12.5700473 24.94466  
## 19 7.42810 0.5911537 14.26505  
## 20 18.75735 12.5700473 24.94466  
## 21 15.71450 9.5493253 21.87967

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



# multiple R-squared value is 0.6794  
#Exponentinal Transformation  
reg\_exp<-lm(log(DT) ~ ST)  
summary(reg\_exp)

##   
## Call:  
## lm(formula = log(DT) ~ ST)  
##   
## Residuals:  
## Min 1Q Median 3Q Max   
## -0.29209 -0.13364 0.02065 0.08421 0.41892   
##   
## Coefficients:  
## Estimate Std. Error t value Pr(>|t|)   
## (Intercept) 2.12137 0.10297 20.601 1.86e-14 \*\*\*  
## ST 0.10555 0.01544 6.836 1.59e-06 \*\*\*  
## ---  
## Signif. codes: 0 '\*\*\*' 0.001 '\*\*' 0.01 '\*' 0.05 '.' 0.1 ' ' 1  
##   
## Residual standard error: 0.1755 on 19 degrees of freedom  
## Multiple R-squared: 0.7109, Adjusted R-squared: 0.6957   
## F-statistic: 46.73 on 1 and 19 DF, p-value: 1.593e-06

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

## 1 2 3 4 5 6 7 8   
## 3.176888 2.543578 2.754681 3.071336 3.176888 2.754681 2.860233 2.438027   
## 9 10 11 12 13 14 15 16   
## 3.176888 3.071336 2.965785 2.543578 2.860233 2.438027 2.438027 2.543578   
## 17 18 19 20 21   
## 2.754681 2.860233 2.332475 2.860233 2.649130

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

## [1] 0.1669628

confint(reg\_exp, level=0.95)

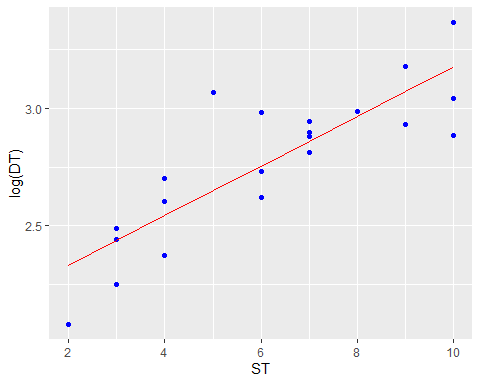
## 2.5 % 97.5 %  
## (Intercept) 1.90584807 2.3368956  
## ST 0.07323457 0.1378686

exp(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 23.97203 16.138575 35.60775  
## 2 12.72512 8.679275 18.65695  
## 3 15.71603 10.789743 22.89153  
## 4 21.57071 14.650800 31.75904  
## 5 23.97203 16.138575 35.60775  
## 6 15.71603 10.789743 22.89153  
## 7 17.46560 11.980604 25.46174  
## 8 11.45042 7.753250 16.91061  
## 9 23.97203 16.138575 35.60775  
## 10 21.57071 14.650800 31.75904  
## 11 19.40993 13.266335 28.39859  
## 12 12.72512 8.679275 18.65695  
## 13 17.46560 11.980604 25.46174  
## 14 11.45042 7.753250 16.91061  
## 15 11.45042 7.753250 16.91061  
## 16 12.72512 8.679275 18.65695  
## 17 15.71603 10.789743 22.89153  
## 18 17.46560 11.980604 25.46174  
## 19 10.30341 6.908812 15.36592  
## 20 17.46560 11.980604 25.46174  
## 21 14.14173 9.690362 20.63787

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



# p value<0.05 and multiple R-squared value is 0.7109  
#polynomial of second degree tronsformation  
reg\_poly<-lm(DT ~ ST+I(ST^2), data=data)  
summary(reg\_poly)

##   
## Call:  
## lm(formula = DT ~ ST + I(ST^2), data = data)  
##   
## Residuals:  
## Min 1Q Median 3Q Max   
## -4.4324 -1.6951 -0.5365 0.9075 6.6676   
##   
## Coefficients:  
## Estimate Std. Error t value Pr(>|t|)   
## (Intercept) 3.5222 4.1597 0.847 0.4082   
## ST 2.8130 1.4608 1.926 0.0701 .  
## I(ST^2) -0.0932 0.1151 -0.810 0.4286   
## ---  
## Signif. codes: 0 '\*\*\*' 0.001 '\*\*' 0.01 '\*' 0.05 '.' 0.1 ' ' 1  
##   
## Residual standard error: 2.962 on 18 degrees of freedom  
## Multiple R-squared: 0.6934, Adjusted R-squared: 0.6594   
## F-statistic: 20.36 on 2 and 18 DF, p-value: 2.391e-05

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

## 1 2 3 4 5 6 7 8   
## 22.332430 13.283069 17.045108 21.290194 22.332430 17.045108 18.646533 11.122455   
## 9 10 11 12 13 14 15 16   
## 22.332430 21.290194 20.061562 13.283069 18.646533 11.122455 11.122455 13.283069   
## 17 18 19 20 21   
## 17.045108 18.646533 8.775444 18.646533 15.257287

confint(reg\_poly, level=0.95)

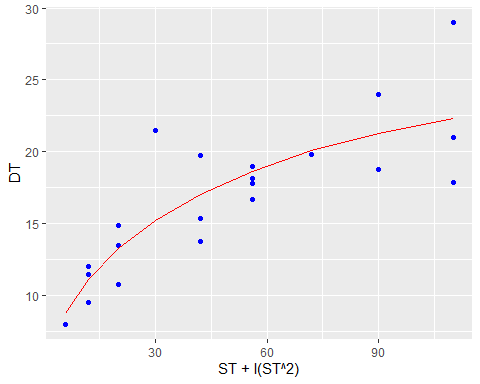
## 2.5 % 97.5 %  
## (Intercept) -5.2169258 12.2613936  
## ST -0.2560669 5.8820703  
## I(ST^2) -0.3349939 0.1485975

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 22.332430 15.360768 29.30409  
## 2 13.283069 6.796484 19.76965  
## 3 17.045108 10.506802 23.58341  
## 4 21.290194 14.728917 27.85147  
## 5 22.332430 15.360768 29.30409  
## 6 17.045108 10.506802 23.58341  
## 7 18.646533 12.120732 25.17233  
## 8 11.122455 4.434281 17.81063  
## 9 22.332430 15.360768 29.30409  
## 10 21.290194 14.728917 27.85147  
## 11 20.061562 13.573170 26.54995  
## 12 13.283069 6.796484 19.76965  
## 13 18.646533 12.120732 25.17233  
## 14 11.122455 4.434281 17.81063  
## 15 11.122455 4.434281 17.81063  
## 16 13.283069 6.796484 19.76965  
## 17 17.045108 10.506802 23.58341  
## 18 18.646533 12.120732 25.17233  
## 19 8.775444 1.423580 16.12731  
## 20 18.646533 12.120732 25.17233  
## 21 15.257287 8.758744 21.75583

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



# multiple R-squared value is 0.6934  
#polynomial of Three degree transformation  
reg\_poly1<-lm(DT ~ ST+I(ST^2)+I(ST^3))  
summary(reg\_poly1)

##   
## Call:  
## lm(formula = DT ~ ST + I(ST^2) + I(ST^3))  
##   
## Residuals:  
## Min 1Q Median 3Q Max   
## -4.8972 -1.7972 -0.1601 0.8077 6.2028   
##   
## Coefficients:  
## Estimate Std. Error t value Pr(>|t|)  
## (Intercept) -4.15818 10.98653 -0.378 0.710  
## ST 7.50248 6.37003 1.178 0.255  
## I(ST^2) -0.92525 1.10553 -0.837 0.414  
## I(ST^3) 0.04446 0.05874 0.757 0.460  
##   
## Residual standard error: 2.998 on 17 degrees of freedom  
## Multiple R-squared: 0.7034, Adjusted R-squared: 0.6511   
## F-statistic: 13.44 on 3 and 17 DF, p-value: 9.586e-05

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

## 1 2 3 4 5 6 7 8   
## 22.797240 13.892856 17.150048 20.827000 22.797240 17.150048 18.270140 11.222284   
## 9 10 11 12 13 14 15 16   
## 22.797240 20.827000 19.406879 13.892856 18.270140 11.222284 11.222284 13.892856   
## 17 18 19 20 21   
## 17.150048 18.270140 7.501412 18.270140 15.779865

confint(reg\_poly1, level=0.95 )

## 2.5 % 97.5 %  
## (Intercept) -27.33772564 19.0213717  
## ST -5.93710786 20.9420660  
## I(ST^2) -3.25771246 1.4072034  
## I(ST^3) -0.07947156 0.1683837

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

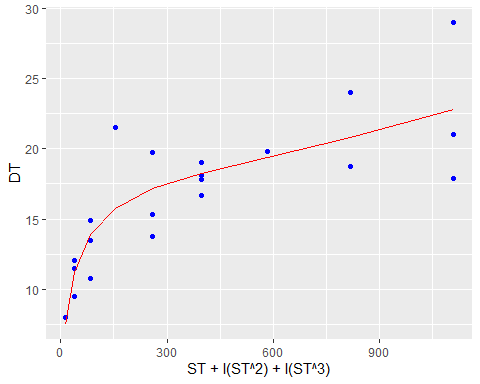
## Warning in predict.lm(reg\_poly1, interval = "predict"): predictions on current data refer to \_future\_ responses

## fit lwr upr  
## 1 22.797240 15.5939839 30.00050  
## 2 13.892856 7.0844910 20.70122  
## 3 17.150048 10.4982995 23.80180  
## 4 20.827000 14.0344853 27.61951  
## 5 22.797240 15.5939839 30.00050  
## 6 17.150048 10.4982995 23.80180  
## 7 18.270140 11.5550572 24.98522  
## 8 11.222284 4.4189573 18.02561  
## 9 22.797240 15.5939839 30.00050  
## 10 20.827000 14.0344853 27.61951  
## 11 19.406879 12.5644227 26.24933  
## 12 13.892856 7.0844910 20.70122  
## 13 18.270140 11.5550572 24.98522  
## 14 11.222284 4.4189573 18.02561  
## 15 11.222284 4.4189573 18.02561  
## 16 13.892856 7.0844910 20.70122  
## 17 17.150048 10.4982995 23.80180  
## 18 18.270140 11.5550572 24.98522  
## 19 7.501412 -0.7718603 15.77468  
## 20 18.270140 11.5550572 24.98522  
## 21 15.779865 9.0162244 22.54351

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

## [1] 2.697085

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



# multiple R-squared value is 0.7034  
#Exponential model gives the best multiple R-squared value is 0.7109 and adjusted r-squared value is 0.6957 and p values<0.05  
predicted\_dt\_values<-exp(predict(reg\_exp))  
predicted\_dt\_values

## 1 2 3 4 5 6 7 8   
## 23.97203 12.72512 15.71603 21.57071 23.97203 15.71603 17.46560 11.45042   
## 9 10 11 12 13 14 15 16   
## 23.97203 21.57071 19.40993 12.72512 17.46560 11.45042 11.45042 12.72512   
## 17 18 19 20 21   
## 15.71603 17.46560 10.30341 17.46560 14.14173

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

