**Delivery Time:**

Here we have to predict delivery time using sorting time.

Output(y) : delivery time

Input(x) : sorting time

**R Code:**

**# Load the library**

library(readr)

**# Load the csv file and stored in object calories\_consumed**

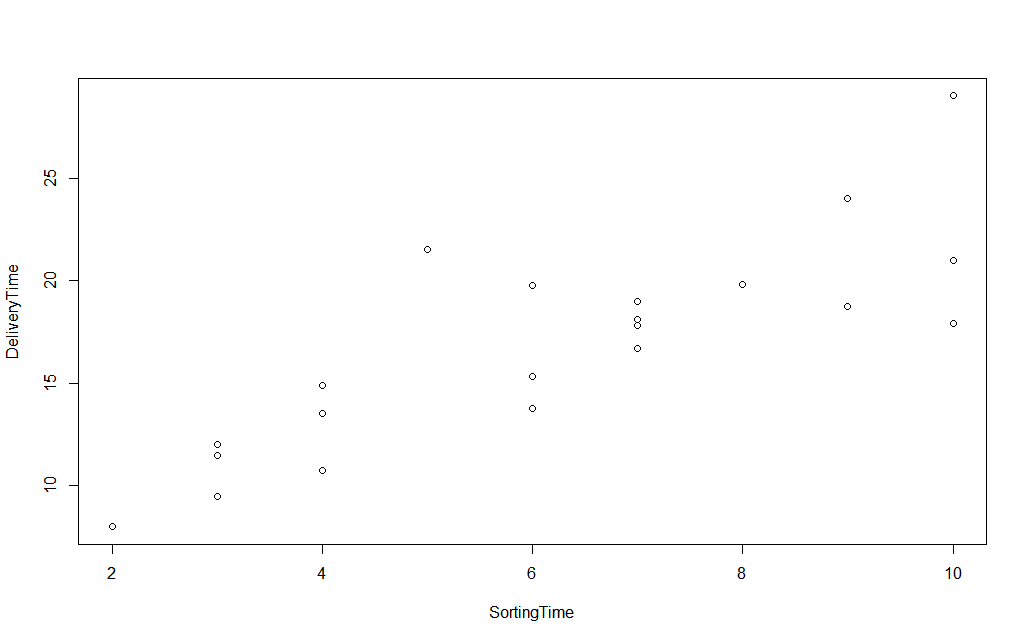
delivery\_time <- read\_csv("D:/ALL Assignments/3.Simple Linear Regression/delivery\_time.csv")

**# attach the object**

attach(delivery\_time)

**# Draw scatter diagram**

plot(SortingTime,DeliveryTime)



It tell following things:

I) Direction : positive correlation

II) Strength : moderate to strong

III) Linearity :Linear relationship

**#Correlation coefficient r :**

cor(SortingTime,DeliveryTime)

It give r = 0.82599

As r between 0.65 - 0.85 => Moderate strength

**#Linear regression technique and its summary**

delivery\_time\_model <- lm(DeliveryTime~SortingTime)

summary(delivery\_time\_model)

It gives:

Call:

lm(formula = DeliveryTime ~ SortingTime)

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 \*\*

SortingTime 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

As we are getting two and three stars(probability of getting wrong is less) and residuals falls the normal distribution but R-squared value is less than 0.8 indicates an overall strength of model is not strong.

Prediction model equation :

**DeliveryTime = 6.5827 + 1.6490(SortingTime)**

**R-squared = 0.68**

**To improve R-squared value we perform some transformation techniques:**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Regression** | **Y** | **X** | **R-squared** | **RMSE** | **Problem** |
| **Linear** | **Y** | **x** | **0.6823** | **2.79** | **Hetroscedasiticity** |
| **Logarithm** | **y** | **Log(x)** | **0.6954** | **2.79** | **hetroscedasiticity** |
| **exponential** | **Log(y)** | **x** | **0.7109** | **2.94** | **No-hetroscedasicity But appears to have non-linear relationship** |
| **Polynomial(2nd degree-quadratic)** | **Log(y)** | **X, x^2** | **0.7649** |  | **We use log in deliveryTime to get rid of hetroscedasicity problem** |
| **3 degree polynomial** | **Log(y)** | **X,x^2,x^3** | **0.7819** |  |  |

From above we see exponential transform give a better model than rest.

Confidence interval estimation:

**confint(delivery\_time\_model\_3,level = 0.95)**

Its gives:

2.5 % 97.5 %

(Intercept) 1.90584807 2.3368956

SortingTime 0.07323457 0.1378686

So,

Lower limit => log(delivery time) = 1.906 + 0.07(Sorting time)

=> delivery time = exp[1.906 + 0.07(Sorting time)]

Upper Limit => log(delivery time) = 2.337+ 0.138(Sorting time)

=> delivery time = exp[2.337 + 0.138(Sorting time)]