**Simple Linear Regression**

1. **Calories\_consumed-> predict weight gained using calories consumed**

**Ans:** calories<-read.csv(file.choose())

View(calories)

attach(calories)

plot(calories$Weight.gained..grams.,calories$Calories.Consumed)

cor(Calories.Consumed,Weight.gained..grams.)

reg<-lm(Calories.Consumed~Weight.gained..grams.)

summary(reg)

pred<-predict(reg)

summary(reg$residuals)

sqrt(sum(reg$residuals)^2/nrow(calories))#RMSE

sqrt(mean(reg$residuals)^2)

confint(reg,level=0.95)

predict(reg,interval="prediction")

ggplot(data=calories,aes(x=Weight.gained..grams.,y=Calories.Consumed)) +

geom\_point(color='red') + geom\_line(color='blue',data=calories,aes(x=Weight.gained..grams.,y=Calories.Consumed))

?ggplot2

Multiple R-squared: 0.8968, Adjusted R-squared: 0.8882

This is best model.no need to go for transformations.

1. **Delivery\_time -> Predict delivery time using sorting time**

**Ans:** delivery\_time<-read.csv(file.choose())

View(delivery\_time)

attach(delivery\_time)

plot(Delivery.Time,Sorting.Time)

cor(Delivery.Time,Sorting.Time)

mod<-lm(Sorting.Time ~ Delivery.Time)

summary(mod)

pred<-predict(mod)

summary(mod$residuals)

sqrt(sum(mod$residuals)^2/nrow(delivery\_time))

sqrt(mean(mod$residuals)^2)

confint(mod,level=0.95)

predict(mod,interval="prediction")

**#logthemic model**

#x=log(deliverytime),y=sorting time

plot(log(Delivery.Time),Sorting.Time)

cor(log(Delivery.Time),Sorting.Time)

mod1<-lm(Sorting.Time~log(Delivery.Time))

summary(mod1)

predict(mod1)

mod1$residuals

sqrt(sum(mod1$residuals^2)/nrow(delivery\_time))#RMSE

sqrt(mean(mod1$residuals^2))

confint(mod1,level = 0.95)

predict(mod1,interval = "confidence")

**#Exponential model**

#x=delivery time y=log(sorting time)

plot(Delivery.Time,log(Sorting.Time))

cor(Delivery.Time,log(Sorting.Time))

mod2<-lm(log(Sorting.Time)~Delivery.Time)

summary(mod2)

mod2$residuals

sqrt(mean(mod2$residuals^2))

logat<-predict(mod2)

at<-exp(logat)

error<-delivery\_time$Sorting.Time-at

error

sqrt(sum(mod2$residuals^2)/nrow((delivery\_time))#RMSE

confint(mod2,level = 0.95)

predict(mod2,interval = "confidence")

**#polynominal model with 2 degree(quadratic model)**

plot(Delivery.Time,Sorting.Time)

plot(Delivery.Time\*Delivery.Time,Sorting.Time)

plot(Delivery.Time\*Delivery.Time,log(Sorting.Time))

cor(Delivery.Time\*Delivery.Time,Sorting.Time)

cor(Delivery.Time\*Delivery.Time,log(Sorting.Time))

mod3<-lm(log(Sorting.Time)~Delivery.Time + I(Delivery.Time\*Delivery.Time))

summary(mod3)

mod3$residuals

logpol<-predict((mod3))

expy<-exp(logpol)

err<-delivery\_time$Sorting.Time - expy

err

sqrt(sum(mod3$residuals^2)/nrow(delivery\_time))

confint(mod3,level=0.95)

predict(mod3,interval = "confidence")

Multiple R-squared: 0.7937, Adjusted R-squared: 0.7708

polynominal model with 2 degree(quadratic model).this is the best model.

**3)Emp\_data -> Build a prediction model for Churn\_out\_rate**

**Ans:**

empdata<-read.csv(file.choose())

View(empdata)

attach(empdata)

plot(Salary\_hike,Churn\_out\_rate)

cor(Salary\_hike,Churn\_out\_rate)

emp1<-lm(Churn\_out\_rate ~ Salary\_hike)

summary(emp1)

pred<-predict(emp1)

summary(emp1$residuals)

sqrt(sum(emp1$residuals)^2/nrow(empdata))

sqrt(mean(emp1$residuals)^2)

confint(emp1,level=0.95)

predict(emp1,interval="prediction")

#logthemic model

#x=log(salary hike),y=Churn\_out\_rate

plot(log(Salary\_hike),Churn\_out\_rate)

cor(log(Salary\_hike),Churn\_out\_rate)

emp2<-lm(Churn\_out\_rate~log(Salary\_hike))

summary(emp2)

predict(emp2)

emp2$residuals

sqrt(sum(emp2$residuals^2)/nrow(empdata))#RMSE

sqrt(mean(emp2$residuals^2))

confint(emp2,level = 0.95)

predict(emp2,interval = "confidence")

#Residual standard error: 4.233 on 8 degrees of freedom

#Multiple R-squared: 0.8486, Adjusted R-squared: 0.8297

#F-statistic: 44.85 on 1 and 8 DF, p-value: 0.0001532

#MR-squred value is near by 0.86.this is the best model

1. **Salary\_hike -> Build a prediction model for Salary\_hike**

**Ans:**  sal\_hike<-read.csv(file.choose())

View(sal\_hike)

attach(sal\_hike)

plot(YearsExperience,Salary)

cor(YearsExperience,Salary)

sal1<-lm(Salary ~ YearsExperience)

summary(sal1)

pred<-predict(sal1)

summary(sal1$residuals)

sqrt(sum(sal1$residuals)^2/nrow(sal\_hike))

sqrt(mean(sal1$residuals)^2)

confint(sal1,level=0.95)

predict(sal1,interval="prediction")

Multiple R-squared: 0.957, Adjusted R-squared: 0.9554

This is the best model.