```
# MLR Model

# Install the required packages

install.packages("UsingR")

library(UsingR)

install.packages("lmtest")

library(lmtest)

install.packages("datarium")

library(datarium)


marketing

head(marketing)

names(marketing)


model_1=lm(sales~youtube+facebook+newspaper,data=marketing)

model_1

summary(model_1)


# Test for significance of regression using ANOVA

f=summary(model_1)$fstatistic

f


f_table=qf(.95, df1=f[2], df2=f[3])

f_table


if(f[1]>f_table)print("reject the null hypothesis that the co-efficient of the predictors is
zero") else print("accept the null hypothesis")


# Test of Individual regression coefficient (beta1)
```

```
teststat=coef(summary(model_1))[2,1]/coef(summary(model_1))[2,2]

teststat

n=nrow(marketing)

n

alpha=0.05


# p values of coefficients

Prob_values=coef(summary(model_1))[, "Pr(>|t|)"]

Prob_values


p_value=2*pt(abs(teststat),n-4,lower.tail=F)

p_value


if(p_value<0.05) print("reject the null hypothesis that the coefficient of the predictor is zero")

  else print("accept the null hypothesis")


# t-test

t_table=qt((alpha/2),n-4,lower.tail = F)

t_table


if(teststat>t_table) print("reject the null hypothesis that the coefficient of the predictor is zero")

   else print("accept the null hypothesis")


# Test for intercept (beta0)

teststat1=coef(summary(model_1))[1,1]/coef(summary(model_1))[1,2]

teststat1
```

```
# p values of coefficients
Prob_values=coef(summary(model_1))[, "Pr(>|t|)"]
Prob_values

p_value=2*pt(abs(teststat1),n-4,lower.tail=F)
p_value

if(p_value<0.05) print("reject the null hypothesis that the intercept is zero") else
  print("accept the null hypothesis")

# t-test
t_table=qt((alpha/2),n-4,lower.tail = F)
t_table

if(teststat1>t_table) print("reject the null hypothesis that the intercept is zero") else
 print("accept the null hypothesis")

# Extract Model Components
obs_val=marketing$sales
obs_val
fitted_val=fitted(model_1)
fitted_val
residuals=resid(model_1)
residuals

# Residual Analysis
# Residuals vs Fitted Plot
plot(fitted_val,residuals,
```

```r
    xlab = "Fitted Values",
    ylab = "Residuals",
    main = "Residuals vs Fitted Values")
abline(h=0,lty=2,col="red")
# Random scatter around zero → model is appropriate
# Funnel shape → heteroscedasticity
# Curved pattern → non linearity


# Breusch-Pagan Test(Homoscedasticity)
bp_test=bptest(model_1)
bp_test
# Decision
# p > 0.05 → homoscedasticity satisfied

# Normal Q-Q Plot
qqnorm(residuals)
qqline(residuals, col = "red")

# Shapiro-Wilk Normality Test
shapiro_test=shapiro.test(residuals)
shapiro_test
# Decision rule
# p>0.05 → residuals are approximately normal
# Since p-value (3.939e-09) < 0.05, we reject H_0
# Statistical inference may be compromised (p-values, confidence intervals)

# Durbin-Watson Test(Independence of Residuals)
```

```r
dw_test=dwtest(model_1)

dw_test

# Conclusion

# The Durbin-Watson statistic will always have a value ranging between 0 and 4.

# A value of 2.0 indicates there is no autocorrelation detected in the sample.

# Values from 0 to less than 2 point to positive autocorrelation.

# Values from 2 to 4 mean negative autocorrelation.


# Cook's Distance (Influential Observations)

# Influential points can distort regression coefficients

# They may Inflate or deflate slope estimates

# Affect hypothesis tests

# Reduce forecast accuracy


plot(cooks.distance(model_1),

   type = "h",

   main = "Cook's Distance")

abline(h = 4/length(residuals), lty = 2, col = "red")

# In a Cook's Distance plot:

# Each vertical line represents one observation

# A horizontal reference line is drawn at 4 / n

# Points above the line deserve further investigation


# Decision

# Di < 1        Observation is not influential

# Di ≥ 1        Observation is highly influential

# Di > 4/n Potentially influential (commonly used rule)
```

```
new_values=data.frame(youtube=122.34,facebook=59.59,newspaper=29.18)

new_values


pred_y=predict(model_1,new_values,interval="predict")

pred_y


conf_y=predict(model_1,new_values,interval="confidence")

conf_y


##########################################################################

# The condition number is the maximum condition index.

# Multicollinearity is present when the VIF is higher than 5 to 10 or

# the condition indices are higher than 10 to 30.


#Check for Multicollinearity in R

#Using Correlation Matrix

install.packages("corrplot")

library("corrplot")

corrplot(cor(marketing),method="number")


#Using VIFs

install.packages("olsrr")

library("olsrr")

ols_vif_tol(model_1)


# Using Eigenvalues and Condition Index

ols_eigen_cindex(model_1)
```