Assignment 3

* Aditya Parey

# Read the data from a text file (adjust the file path and format as needed)

> house.data <- read.table("HouseReg.txt")

>

> # Check the column names in the dataset

> names(house.data)

[1] "price" "size"

>

> # Linear regression

> reg.model <- lm(price ~ size, data = house.data)

> summary(reg.model)

Call:

lm(formula = price ~ size, data = house.data)

Residuals:

Min 1Q Median 3Q Max

-11214 -4\>……………………………………………………………413 121 3838 11747

Coefficients:

Estimate Std. Error t value Pr(>|t|)

(Intercept) 209079.77 8330.33 25.099 < 2e-16 \*\*\*

size 30.78 4.69 6.564 9.63e-08 \*\*\*

---

Signif. codes: 0 ‘\*\*\*’ 0.001 ‘\*\*’ 0.01 ‘\*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Residual standard error: 5743 on 38 degrees of freedom

Multiple R-squared: 0.5313, Adjusted R-squared: 0.519

F-statistic: 43.08 on 1 and 38 DF, p-value: 9.629e-08

The regression line would be

Y = 30.78(size) + 209079.77

1. We take three assumptions into factor: normality, constant error variance, mean of residuals is zero, and independence.

Normality:

The Shapiro-Wilk test (as shown in the code) and the Q-Q plot are used to check the normality of residuals. The Shapiro-Wilk test's p-value (0.8356) and the Q-Q plot suggest that the residuals are approximately normally distributed, which is a good sign.

# Normality tests for residuals

resid(reg.model)

shapiro.test(reg.model$residuals)

qqnorm(resid(reg.model))

qqline(resid(reg.model))

A graph of a normal q-q plot

Description automatically generated

Constant error variance and mean of residuals:

The variability of the residuals needs to be constant.

This assumption can be elevated using a plot of the residuals versus the predicted values.

We have a residuals vs fitted plot for this.

By this graph we can see that there are points exactly at zero and near zero, which states that there’s a constant error variance.

This assumption can be evaluated using a plot of the residuals versus the predicted values.

If the residuals are centred around the line at 0 that is evidence, the assumption is valid.

A graph with red lines and numbers

Description automatically generated

Independence:

The residuals need to be independent of each other. One way to evaluate independency that occurs related to the order the data was collected.

A diagram of a number of individuals

Description automatically generated with medium confidence

# Get the residuals from the model

residuals <- resid(reg.model)

# Create a sequence of numbers representing observation order

observation\_order <- 1:length(residuals)

# Create a scatterplot of residuals vs observation order

plot(observation\_order, residuals, xlab = "Observation Order", ylab = "Residuals", main = "Residuals vs Observation Order")

abline(h = 0, col = "red")

1. Scatter plot of the data with the regression line overlaid on it

A graph with dots and lines

Description automatically generated

1. The observations with indexes 5, 16, and 26 are identified as potentially influential.

A screenshot of a computer code

Description automatically generated

1. Hypotheses:

Here, B1 is beta value.

Null Hypothesis (Ho):

H0: B1 = 0

In this equation, B1 represents the coefficient for the explanatory variable (size). The null hypothesis states that there is no significant linear relationship between the response variable (price) and the explanatory variable (size). If B1 is equal to 0, it implies no linear relationship.

Alternative Hypothesis (Ha):

Ha: B1 =/= 0

The alternative hypothesis, on the other hand, states that there is a significant linear relationship between the response variable (price) and the explanatory variable (size). If B1 is not equal to 0, it suggests a linear relationship.

The t-test for the size coefficient in the summary output provides the test statistic (t-value) and the p-value (approx. 0.00000).

We can reject the null hypothesis since the p-value is much smaller than alpha value of 0.05 and we have significant evidence to conclude that there is a significant linear relationship between house size and price.

t-stat value is 43.08

R2adj. = 0.519

getwd()

house.data <- read.table("HW4HouseData.txt", header = TRUE)

names(house.data)

## least-squares regression model ##

Regmodel.house <- lm(Price ~ HouseSize + LotSize + Pool, data = house.data)

summary(Regmodel.house)

shapiro.test(Regmodel.house$residuals)

qqnorm(resid(Regmodel.house))

qqline(resid(Regmodel.house))

par(mfrow = c(2,2))

plot(Regmodel.house)

residuals <- resid(Regmodel.house)

# Create a sequence of numbers representing observation order

observation\_order <- 1:length(residuals)

# Create a scatterplot of residuals vs observation order

plot(observation\_order, residuals, xlab = "Observation Order", ylab = "Residuals", main = "Residuals vs Observation Order")

abline(h = 0, col = "red")

summary(Regmodel.house)$coefficients["HouseSize", c("t value", "Pr(>|t|)")]

summary(Regmodel.house)$coefficients["LotSize", c("t value", "Pr(>|t|)")]

summary(Regmodel.house)$coefficients["PoolYes", c("t value", "Pr(>|t|)")]

new\_data <- data.frame(HouseSize = 2075, LotSize = 19450, Pool = "Yes")

predicted\_price <- predict(Regmodel.house, newdata = new\_data)

predicted\_price

predict(Regmodel.house, newdata = new\_data, interval = "confidence", level = 0.90)

intervals <- confint(Regmodel.house, level = 0.90)

intervals

## Q2 ##

q2.data <- read.table("HW4Q2Data.txt", header = TRUE)

names(q2.data)

model <- lm(y ~ x1 + x2 + x3 + x4 + x5, data = q2.data)

summary(model)

Regmodel.house <- lm(y~ x1 + x2 - x3 - x4 + x5, data = q2.data)

summary(model)

shapiro.test(model$residuals)

qqnorm(resid(model))

qqline(resid(model))

par(mfrow = c(2,2))

plot(model)

residuals <- resid(model)

# Create a sequence of numbers representing observation order

observation\_order <- 1:length(residuals)

# Create a scatterplot of residuals vs observation order

plot(observation\_order, residuals, xlab = "Observation Order", ylab = "Residuals", main = "Residuals vs Observation Order")

abline(h = 0, col = "red")

# full regression model

model <- lm(y ~ x1 + x2 + x3 + x4 + x5, data = q2.data)

# summary of the model

model\_summary <- summary(model)

f\_statistic <- model\_summary$fstatistic[1]

p\_value <- pf(f\_statistic, df1 = model\_summary$fstatistic[2], df2 = model\_summary$fstatistic[3], lower.tail = FALSE)

f\_statistic

p\_value

##library(MASS)

# Start with an intercept-only model

MultiReg.empty <- lm(y ~ 1, data = q2.data)

stepForward1 = add1(MultiReg.empty, scope = q2.data, test = "F", trace=TRUE)

stepForward1

#x1 has the highest significant F-value

MultiReg.empty2 <- lm(y ~ x1, data = q2.data)

stepForward2 = add1(MultiReg.empty2, scope = q2.data, test = "F", trace=TRUE)

stepForward2

#x2 has the highest significant F-value

MultiReg.empty3 <- lm(y ~ x1+x2, data = q2.data)

stepForward3 = add1(MultiReg.empty3, scope = q2.data, test = "F", trace=TRUE)

stepForward3

#x5 has the highest significant F-value

MultiReg.empty4 <- lm(y ~ x1+x2+x5, data = q2.data)

stepForward4 = add1(MultiReg.empty4, scope = q2.data, test = "F", trace=TRUE)

stepForward4

shapiro.test(MultiReg.empty4$residuals)

qqnorm(resid(MultiReg.empty4))

qqline(resid(MultiReg.empty4))

plot(MultiReg.empty4$fitted.values,rstandard(MultiReg.empty4))

par(mfrow=c(2,2))

plot(MultiReg.empty4)

par(mfrow=c(1,1))

##library(car)

vif(MultiReg.empty4)

coefficients <- coef(MultiReg.empty4)

coefficients

#backward elimination

# Fit a full regression model with all predictors

MultiReg.full <- lm(y ~ x1 + x2 + x3 + x4 + x5, data = q2.data)

# Perform backward selection to drop the least significant predictors (x3 and x4)

stepBack1 <- drop1(MultiReg.full, scope = ~ x1 + x2 + x3 + x4 + x5, test = "F", trace = TRUE)

stepBack1

# Remove the least significant predictors (x3 and x4) from the linear model

MultiReg.full2 <- lm(y ~ x1 + x2 + x5, data = q2.data)

stepBack2 <- drop1(MultiReg.full2, scope = ~ x1 + x2 + x5, test = "F", trace = TRUE)

stepBack2

vif(MultiReg.full2)

coefficients2 <- coef(MultiReg.full2)

coefficients2

# this is essentially the same model as the forward selection so we don't

# need to verify the assumptions

#stepwise elimination

##chooseCRANmirror(graphics = FALSE)

##install.packages("Matrix", type = "binary")

##install.packages("MatrixModels", type = "binary")

##install.packages("rms")

##library(rms)

##

MultiReg.stepwise=ols(y~x1+x2+x3+x4+x5,data=q2.data)

stepStepwise=fastbw(MultiReg.stepwise,rule="p")

stepStepwise

MultiReg.stepwise2=lm(y~x2+x5,data=q2.data)

summary(MultiReg.stepwise2)

shapiro.test(resid(MultiReg.stepwise2))

par(mfrow=c(2,2))

plot(MultiReg.stepwise2)

par(mfrow=c(1,1))

##library(car)

vif(MultiReg.stepwise2)

coefficients3 <- coef(MultiReg.stepwise2)

coefficients3

A screenshot of a computer

Description automatically generated

1. The 90% confidence interval for the average price of a house with 1730 square feet is between the values 260,775.4 to 263,888.8.

A close-up of blue text

Description automatically generated

1. The 98% confidence interval for the price of a randomly selected house with 1730 square feet is 248206.6 to 276457.5

A close-up of blue text

Description automatically generated

1. The 95% confidence interval for the slope coefficient (the relationship between house size and price) is between 21.28803 to 40.27531. This means that we are 95% confident that the true population slope between house size and price falls within this interval.

A close-up of numbers

Description automatically generated

Full code to get these answers:

# Set the working directory to the location of your data file

getwd()

# Read the data from a text file (adjust the file path and format as needed)

house.data <- read.table("HouseReg.txt")

# Check the column names in the dataset

names(house.data)

# Linear regression

reg.model <- lm(price ~ size, data = house.data)

summary(reg.model)

# Load the "psych" library for descriptive statistics (if not already loaded)

library("psych")

# Descriptive statistics for "price" and "size"

describe(house.data$price)

describe(house.data$size)

# Calculate the estimated slope coefficient (est.b1)

est.b1 <- cor(house.data$price, house.data$size) \* sd(house.data$price) / sd(house.data$size)

est.b1

# Calculate the estimated intercept (est.b0)

est.b0 <- mean(house.data$price) - est.b1 \* mean(house.data$size)

est.b0

# Create a scatter plot of price against size with the regression line

plot(house.data$size, house.data$price)

abline(reg.model, col = "red")

# Normality tests for residuals

resid(reg.model)

shapiro.test(reg.model$residuals)

qqnorm(resid(reg.model))

qqline(resid(reg.model))

#Residual vs Predicted graphs

residuals <- resid(reg.model)

predicted\_values <- predict(reg.model)

# Create the residuals vs. predicted values plot

plot(predicted\_values, residuals, xlab = "Predicted Values", ylab = "Residuals", main = "Residuals vs Predicted Values")

abline(h = 0, col = "red")

#Independence

# Get the residuals from the model

residuals <- resid(reg.model)

# Create a sequence of numbers representing observation order

observation\_order <- 1:length(residuals)

# Create a scatterplot of residuals vs observation order

plot(observation\_order, residuals, xlab = "Observation Order", ylab = "Residuals", main = "Residuals vs Observation Order")

abline(h = 0, col = "red")

# Diagnostic plots

plot(reg.model$fitted.values, rstandard(reg.model))

# Set up a 2x2 grid for diagnostic plots

par(mfrow = c(2, 2))

plot(reg.model)

# Influence measures

infl <- influence.measures(reg.model)

summary(infl)

# Summary of the regression model

summary(reg.model)

# Confidence intervals for coefficients

confint(reg.model, level = 0.95)

# Fit the regression model again (this step is redundant, as you've already fitted the model)

reg.model <- lm(price ~ size)

# Create a data frame for prediction

prediction.value <- data.frame(size = 1730)

# Predict the price for a house with 1730 square feet

predict(reg.model, prediction.value)

# Predict with a 90% confidence interval

predict(reg.model, prediction.value, interval = "confidence", level = 0.90)

# Predict with a 98% prediction interval

predict(reg.model, prediction.value, interval = "prediction", level = 0.98)

# Predict with a 95% prediction interval

predict(reg.model, prediction.value, interval = "prediction", level = 0.95)

# Confidence intervals for coefficients

confint(reg.model)