Load the dataset auto.csv (given in the shared Google folder) into R. This dataset has some columns:

**name**: name of the car model (e.g., bmw 2002)

**origin**: where the car is produced. 1 = US, 2 = EU, 3 = Asia.

**mpg**: miles per gallon (the higher the better).

**weight**: weight in lbs

**model\_year**: the model year

**horsepower**: the engine power (measured in horsepower). The higher the stronger.

**cylinders**: the number of cylinders in the car engine. The higher the stronger.

**acceleration**: the time (in seconds) for the car to speed up from 0 to 60 mph.

**displacement**: the volume of air the car engine can take in to burn.

To answer the below questions, we need to set the directory where the “auto.csv” file is located.

In my local machine, the code to set the working directory is

setwd(“C:/Users/cscha/OneDrive/Desktop/Software analytics”)

Now read the “auto.csv” file as typing the following code

> cars <- read.csv(“auto.csv”)

Here, “cars” work as a variable to perform operations on the “auto.csv” file

Q1. Build a linear regression model M1 to estimate **horsepower** from **cylinders.** Explain the parameters of this model. If you know a car having 10 cylinders; what is its estimated horsepower?

The ‘R’ code to build a linear regression model M1 is as follows:

>M1\_model <- lm (horsepower ~ cylinders, data = cars)

>summary(M1\_model)

Here the horsepower is the dependent variable, and the cylinders are the independent variable.

The variable we are attempting to predict or explain is referred to as the dependent variable, also known as the response variable or target variable. It reflects the result or the relevant variable in the investigation at hand. To model and comprehend how changes in the independent variables affect the dependent variable is the aim of a linear regression model. Because we are attempting to forecast horsepower in the current scenario using other variables like cylinders, horsepower is the dependent variable.

The factors used to explain or predict the variance in the dependent variable are referred to as independent variables, predictor variables, or features. These variables are applied to a model of the dependent variable's relationship. Because it is utilized to forecast horsepower in the scenario, the number of cylinders is the independent variable.

In the above code

M1\_model <- lm (horsepower ~ cylinders, data = cars), builds the linear regression model based on the formula

horsepower ~ cylinders which are stored in cars data. The resultant data is stored in M1\_model.

>summary(M1\_model) it gives the following output model,

Call:

lm(formula = horsepower ~ cylinders, data = cars)

Residuals:

Min 1Q Median 3Q Max

-62.558 -12.558 -2.558 11.530 77.442

Coefficients:

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

(Intercept) 0.3822 3.5226 0.108 0.914

cylinders 19.0220 0.6147 30.947 <2e-16 \*\*\*

---

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

Residual standard error: 20.73 on 390 degrees of freedom

Multiple R-squared: 0.7106, Adjusted R-squared: 0.7099

F-statistic: 957.7 on 1 and 390 DF, p-value: < 2.2e-16

From the above output, we can state that

Call represents the formula that was used to build the linear regression model.

In Residuals, we can see the discrepancies between the actual horsepower levels and the model-predicted values. They are employed to judge how well the model fits the provided data.

Min: Since there is a minimal residual of -62.558, the model has some room for error. The largest residual is 77.442, which shows that the model has a 7% margin of error for overestimating horsepower.

From the coefficients section, we can say that the intercept is 0.382 whereas the estimated coefficient for cylinders is 19.022. This means that this amount of value increase in horsepower is predicted for every new cylinder.

The average difference between the observed horsepower values and the model's anticipated values is measured by the residual standard error. In this instance, it's roughly 20.73.

The overall significance of the model is evaluated using the F-statistic. In this instance, it is 957.7 with an extremely low p-value ( 2.2e-16), demonstrating the statistical significance of the model.

In conclusion, this output offers details on the model's coefficients, its accuracy, and its overall importance. It implies that the quantity of cylinders is a highly significant predictor of horsepower, and the model accounts for a sizable proportion of the variability in horsepower.

The ‘R’ code to calculate the horsepower having 10 cylinders is as follows:

>required\_data <- data.frame(cylinders=10)

> horsepower\_predictor-<-predict (M1\_model , new\_data = required\_data)

>summary(horsepower\_predcition)

In the above code required\_data <- data. frame(cylinders=10), creates a new data frame with one column named as cylinders, and sets its value to 10. With the help of the data in this data frame, we may estimate the horsepower of a car with ten cylinders. Based on the M1\_model (linear regression model), this line uses the predict function to make predictions about horsepower.

The formula newdata = required\_data designates the new data frame (required\_data) for which predictions are desired.

The output is:

Min. 1st Qu. Median Mean 3rd Qu. Max.

57.45 76.47 76.47 104.47 152.56 152.56

The linear regression model M1\_model predicts that a car with 10 cylinders will have roughly 152.56 horsepower as the maximum value.

Q2. Build a linear regression model M2 to estimate **horsepower** from **displacement.** Explain the parameters of this model. If you know a car having an engine displacement of 200; what is its estimated horsepower?

The ‘R’ code to build a linear regression model M2 is as follows:

>M2\_model <- lm (horsepower ~ displacement , data = cars)

>summary(M2\_model)

Here the horsepower is the dependent variable, and the displacement is the independent variable.

The variable we are attempting to predict or explain is referred to as the dependent variable, also known as the response variable or target variable. It reflects the result or the relevant variable in the investigation at hand. To model and comprehend how changes in the independent variables affect the dependent variable is the aim of a linear regression model. Because we are attempting to forecast horsepower in the current scenario using other variables like displacement, horsepower is the dependent variable.

The factors used to explain or predict the variance in the dependent variable are referred to as independent variables, predictor variables, or features. These variables are applied to a model of the dependent variable's relationship. Because it is utilized to forecast horsepower in the scenario, the displacement is the independent variable.

In the above code

M2\_model <- lm (horsepower ~ displacement, data = cars), builds the linear regression model based on the formula

horsepower ~ displacement which is stored in cars data. The resultant data is stored in M2\_model.

>summary(M2\_model) it gives the following output model,

Call:

lm(formula = horsepower ~ displacement, data = cars)

Residuals:

Min 1Q Median 3Q Max

-50.819 -10.695 -0.819 8.676 64.742

Coefficients:

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

(Intercept) 40.306108 1.815099 22.21 <2e-16 \*\*\*

displacement 0.330038 0.008223 40.13 <2e-16 \*\*\*

---

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

Residual standard error: 17.02 on 390 degrees of freedom

Multiple R-squared: 0.8051, Adjusted R-squared: 0.8046

F-statistic: 1611 on 1 and 390 DF, p-value: < 2.2e-16

From the above output we can state that,

Call represents the formula that was used to build the linear regression model.

In Residuals, we can see the discrepancies between the actual horsepower levels and the model-predicted values. They are employed to judge how well the model fits the provided data.

Min: Since there is a minimal residual of -50.819, the model has some room for error. The largest residual is 64.742, which shows that the model can overestimate horsepower.

From the coefficients section, we can say that the intercept is 40.306 whereas the estimated coefficient for displacement is 0.33. This indicates that horsepower is anticipated to rise by about 0.330038 units for every increased unit of engine displacement.

The average difference between the observed horsepower values and the model's anticipated values is measured by the residual standard error. In this instance, it's roughly 17.02.

The overall significance of the model is evaluated using the F-statistic. In this instance, it is 1611 with an extremely low p-value ( 2.2e-16), demonstrating the statistical significance of the model.

In conclusion, this output offers details on the model's coefficients, its accuracy, and its overall importance. The model accounts for a sizable amount of variability in horsepower, indicating that engine displacement is a highly relevant predictor of horsepower.

The ‘R’ code to calculate the horsepower having an engine displacement as 200 is as follows:

>required\_data <- data.frame(displacement = 200)

> horsepower\_predictor <- predict (M2\_model , new\_data = required\_data)

>summary(horsepower\_predcitor)

In the above code required\_data <- data. frame(displacement = 200), creates a new data frame with one column named as displacement, and sets its value to 200. With the help of the data in this data frame, we may estimate the horsepower of a car with an engine displacement of 200. Based on the M2\_model (linear regression model), this line uses the predict function to make predictions about horsepower.

The formula newdata = required\_data designates the new data frame (required\_data) for which predictions are desired.

The output is:

Min. 1st Qu. Median Mean 3rd Qu. Max.

62.75 74.96 90.14 104.47 131.31 190.47

The linear regression model M2\_model predicts that a car with an engine displacement of 200 will have roughly 190.47 horsepower as its maximum value.

Q3. Compare M1 and M2. What model has better goodness of fit?

Extract 20 samples from the dataset and estimate **horsepower** using both M1 and M2 on those samples. Compare the errors using a paired t.test and explain the result

We can start by comparing the respective R-squared values of models M1 and M2 to see whether the model has a greater goodness of fit. R-squared offers a consistent and comprehensible measurement of how well a linear regression model explains the variation in the dependent variable, making it an important statistic for comparing goodness of fit.

The ‘R’ code to compare M1 and M2 models for best goodness of fit is as follows:

>summary(M1\_model) $r. squared

The output is:

[1] 0.7106209

>summary(M2\_model)$r. squared

The output is:

[1] 0.8050701

From the above outputs we can state that model M2 has a better goodness of fit to the data.

The ‘R’ code to extract 20 samples from the dataset and estimate horsepower using m1 and m2 models is as follows:

>set.seed(120)

>required\_indices <- sample(nrow(cars) ,20)

>required\_data <- cars [required\_indices , ]

>M1\_horsepower\_predictor < -predict (M1\_model, new\_data = required\_data)

>M2\_horsepower\_predictor <- predict (M2\_model , new\_data=required\_data)

>summary(M1\_model\_horsepower\_predictor)

The output is :

Min. 1st Qu. Median Mean 3rd Qu. Max.

57.45 76.47 76.47 104.47 152.56 152.56

>summary(M2\_model\_horsepower\_predictor)

The output is:

Min. 1st Qu. Median Mean 3rd Qu. Max.

62.75 74.96 90.14 104.47 131.31 190.47

From the above ‘R’ code we can state that

required\_indices <- sample(nrow(cars), 20), it performs the sample function where it collected some 20 random sample row indices from the cars dataset.

required\_data <- cars [required\_indices , ] is used to subset the cars dataset which includes all the columns based on the random indices stored in the required\_indices vector.

M1\_horsepower\_predictor <- predict (M1\_model, new\_data=required\_data), In essence, the line of code uses the M1\_model to forecast "horsepower" values for a certain set of data that is provided in required\_data. You can use the M1\_horsepower\_predictor variable to save the predicted values and use them for further analysis or evaluation.

M2\_horsepower\_predictor <- predict (M2\_model, new\_data=required\_data), In essence, the line of code uses the M2\_model to forecast "horsepower" values for a certain set of data that is provided in required\_data. You can use the M2\_horsepower\_predictor variable to save the predicted values and use them for further analysis or evaluation.

Summary prints the summary of horsepower\_predictor.

The ‘R’ code to compare the errors is as follows:

> M1\_errors <- M1\_horsepower\_predictor – required\_data$horsepower

> M2\_errors <- M2\_horsepower\_predictor - required\_data$horsepower

>summary(M1\_errors)

>summary(M2\_errors)

> t\_test <- t.test (M1\_errors, M2\_errors, paired=TRUE)

>print(t\_test)

The output of summary(M1\_errors) is:

Min. 1st Qu. Median Mean 3rd Qu. Max.

-143.5298 -28.5298 0.4702 1.6786 35.5142 92.5582

The output of summary(M2\_errors) is:

Min. 1st Qu. Median Mean 3rd Qu. Max.

-153.621 -24.071 -1.924 1.679 34.976 112.321

The output of paired t-test is:

Paired t-test

data: M1\_errors and M2\_errors

t = 1.1537e-12, df = 391, p-value = 1

alternative hypothesis: true mean difference is not equal to 0

95 percent confidence interval:

-1.062937 1.062937

sample estimates:

mean difference

6.237562e-13

The observed values of the variable horsepower in the dataset required\_data are subtracted from the predictions made by the two separate models, M1\_horsepower\_predictor and M2\_horsepower\_predictor, to produce M1\_errors and M2\_errors, respectively. These errors show how far each model's predictions from the actual numbers differ.

The output of a paired t-test on the variables M1\_errors and M2\_errors is stored in the variable t\_test. This is a paired t-test, which means that the two sets of errors are related or coupled in some way (for example, they come from the same observations or samples). This is indicated by the paired = TRUE parameter.

The high p-value which is 1 in this case and the confidence interval that includes zero, taken together, show that we are unable to rule out the null hypothesis and that there is no discernible difference between the means of M1\_errors and M2\_errors. In terms of forecasting the variable horsepower in the dataset, these two models seem to perform similarly.

Q4. Build a linear regression model M3 to estimate **mpg** from **cylinders.** Explain the parameters of this model. If you know a car having 10 cylinders; what is its estimated mpg?

The ‘R’ code to build a linear regression model M3 is as follows:

>M3\_model <- lm (mpg ~ cylinders, data = cars)

>summary(M3\_model)

Here the mpg is the dependent variable, and the cylinders are the independent variable.

The variable we are attempting to predict or explain is referred to as the dependent variable, also known as the response variable or target variable. It reflects the result or the relevant variable in the investigation at hand. To model and comprehend how changes in the independent variables affect the dependent variable is the aim of a linear regression model. Because we are attempting to forecast mpg in the current scenario using other variables like cylinders, mpg is the dependent variable.

The factors used to explain or predict the variance in the dependent variable are referred to as independent variables, predictor variables, or features. These variables are applied to a model of the dependent variable's relationship. Because it is utilized to forecast mpg in the scenario, the number of cylinders is the independent variable.

In the above code

M3\_model <- lm (mpg ~ cylinders, data = cars), builds the linear regression model based on the formula

Mpg ~ cylinders which are stored in cars data. The resultant data is stored in M3\_model.

>summary(M3\_model) it gives the following output model,

Call:

lm(formula = mpg ~ cylinders, data = cars)

Residuals:

Min 1Q Median 3Q Max

-14.2413 -3.1832 -0.6332 2.5491 17.9168

Coefficients:

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

(Intercept) 42.9155 0.8349 51.40 <2e-16 \*\*\*

cylinders -3.5581 0.1457 -24.43 <2e-16 \*\*\*

---

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

Residual standard error: 4.914 on 390 degrees of freedom

Multiple R-squared: 0.6047, Adjusted R-squared: 0.6037

F-statistic: 596.6 on 1 and 390 DF, p-value: < 2.2e-16

From the above output, we can state that

Call represents the formula that was used to build the linear regression model.

In Residuals, we can see the discrepancies between the actual mpg levels and the model-predicted values. They are employed to judge how well the model fits the provided data.

Min: Since there is a minimal residual of -14.24, the model has some room for error. The largest residual is 17.9.

From the coefficients section, we can say that the interception of the model is 42.9 and the cylinder variable's coefficient is roughly -3.5581. The projected change in mpg for a one-unit change in cylinders is represented by this coefficient. In this instance, it means that the anticipated miles per gallon (mpg) drops by around 3.5581 for every additional cylinder in the car's engine.

The average difference between the observed mpg values and the model's anticipated values is measured by the residual standard error. In this instance, it's roughly 4.9.

The overall significance of the model is evaluated using the F-statistic. In this instance, it is 596.6 with an extremely low p-value ( 2.2e-16), demonstrating the statistical significance of the model.

In conclusion, this output shows a basic linear regression model that forecasts miles per gallon (mpg) depending on the quantity of cylinders in an automobile's engine. The model appears to be highly significant and accounts for approximately 60.47 percent of the mpg variability. The negative cylinder coefficient implies that the predicted mpg declines as the cylinder count rises.

The ‘R’ code to calculate the mpg having 10 cylinders is as follows:

>required\_data <- data.frame(cylinders=10)

> mpg\_predictor-<-predict (M3\_model , new\_data = required\_data)

>summary(mpg\_predcition)

In the above code required\_data <- data. frame(cylinders=10), creates a new data frame with one column named as cylinders, and sets its value to 10. With the help of the data in this data frame, we may estimate the mpgof a car with ten cylinders. Based on the M3\_model (linear regression model), this line uses the predict function to make predictions about mpg.

The formula newdata = required\_data designates the new data frame (required\_data) for which predictions are desired.

The output is:

. Min. 1st Qu. Median Mean 3rd Qu. Max.

14.45 14.45 28.68 23.45 28.68 32.24

The linear regression model M3\_model predicts that a car with 10 cylinders will have roughly 32.24 mpg as the maximum value.

Q5. Build a linear regression model M4 to estimate **mpg** from **weight.** Explain the parameters of this model. Is M4 better than M3? If yes, why using weight of cars to predict mpg is better than using engine size?

The ‘R’ code to build a linear regression model M3 is as follows:

>M4\_model <- lm (mpg ~ weight, data = cars)

>summary(M3\_model)

Here the mpg is the dependent variable, and the weight is the independent variable.

The variable we are attempting to predict or explain is referred to as the dependent variable, also known as the response variable or target variable. It reflects the result or the relevant variable in the investigation at hand. To model and comprehend how changes in the independent variables affect the dependent variable is the aim of a linear regression model. Because we are attempting to forecast mpg in the current scenario using other variables like weight, mpg is the dependent variable.

The factors used to explain or predict the variance in the dependent variable are referred to as independent variables, predictor variables, or features. These variables are applied to a model of the dependent variable's relationship. Because it is utilized to forecast mpg in the scenario, the weight is the independent variable.

In the above code

M4\_model <- lm (mpg ~ weight, data = cars), builds the linear regression model based on the formula

Mpg ~ weight which are stored in cars data. The resultant data is stored in M4\_model.

>summary(M4\_model) it gives the following output model,

Call:

lm(formula = mpg ~ weight, data = cars)

Residuals:

Min 1Q Median 3Q Max

-11.9736 -2.7556 -0.3358 2.1379 16.5194

Coefficients:

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

(Intercept) 46.216524 0.798673 57.87 <2e-16 \*\*\*

weight -0.007647 0.000258 -29.64 <2e-16 \*\*\*

---

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

Residual standard error: 4.333 on 390 degrees of freedom

Multiple R-squared: 0.6926, Adjusted R-squared: 0.6918

F-statistic: 878.8 on 1 and 390 DF, p-value: < 2.2e-16

From the above output, we can state that

Call represents the formula that was used to build the linear regression model.

In Residuals, we can see the discrepancies between the actual mpg levels and the model-predicted values. They are employed to judge how well the model fits the provided data.

Min: Since there is a minimal residual of -11.97, the model has some room for error. The largest residual is 16.5.

From the coefficients section, we can say that the interception of the model is 46.2 and the weight variable's coefficient is roughly -0.07. The projected change in mpg for a one-unit change in weight is represented by this coefficient. In this instance, it means that the anticipated miles per gallon (mpg) will drop by about 0.0076 for each additional unit of weight.

The average difference between the observed mpg values and the model's anticipated values is measured by the residual standard error. In this instance, it's roughly 4.3.

The overall significance of the model is evaluated using the F-statistic. In this instance, it is 878.8 with an extremely low p-value ( 2.2e-16), demonstrating the statistical significance of the model.

In conclusion, this output shows a linear regression model that forecasts miles per gallon (mpg) using the weight of the cars. The model, which is highly significant, accounts for 69.26% of the variation in mpg. The negative weight coefficient implies that as the car's weight rises, the projected mpg falls.

To conclude which is the best model we first focus on the adjusted r-squared values and residual standard errors.

R-squared is modified based on the number of predictors in the model with adjusted R-squared. Inclusion of pointless predictors is penalized. An adjusted R-squared score that is closer to 1 indicates that the model's predictors are more accurately accounting for a bigger part of the variance in the dependent variable. As a result, a model that has a higher adjusted R-squared has a superior explanatory power to model complexity ratio.

The residual standard error quantifies the typical size of the residuals (mistakes) that the model made in predicting the dependent variable. RSE values below a certain threshold imply a better fit between the model's predictions and the actual observed values.

A lower RSE indicates more accurate predictions from the model and less unexplained variation in the data. On the other hand, a greater RSE signifies a worse model fit because it shows that the model's predictions are less accurate and that there is more unexplained variation.

In conclusion, it is preferable to have larger adjusted R-squared values and lower residual standard errors since they show that the model is better able to capture the underlying patterns and relationships in the data.

The ‘R’ code to calculate the adjusted\_r\_squared values is as follows:

> M3\_adjusted\_r\_squared <- summary(M3\_model)$adj.r.squared

> M4\_adjusted\_r\_squared <- summary(M4\_model)$adj.r.squared

>print(M3\_adjusted\_r\_squared)

>print(M4\_adjusted\_r\_squared)

The output of M3\_adjusted\_r\_squared is:

[1] 0.6036754

The output of M4\_adjusted\_r\_squared is:

[1] 0.6918423

The ‘R’ code to calculate the residual standard errors is as follows:

> M3\_r\_standard\_error <- summary(M3\_model)$sigma

> M4\_r\_standard\_error <- summary(M4\_model)$sigma

>print(M3\_r\_standard\_error)

>print M4\_r\_standard\_error ()

The output of M3\_r\_standard\_error is:

[1] 4.913589

The output of M3\_r\_standard\_error is:

[1] 4.332712

From the above outputs, we can state that M4\_model has a higher adjusted R-squared value and a lower residual standard error which generally indicates a better fit to the data, and M4\_model which has weight as an independent variable is also considered better for predicting mpg-based on the available data. In non-technical terms when compared to engine size, a car's weight can be a more meaningful and logical predictor of its fuel economy (mpg). Cars that are heavier frequently use more energy to run, which might reduce their fuel efficiency. Due to the alignment of these two variables with accepted wisdom, weight becomes a more interpretable predictor.