APPLIED STATISTICAL ANALYSIS LAB

NAME: ADITI KULKARNI

ROLL NO: 55

YEAR: SY

DIVISION: E (E3)

SRN NO: 202201893

ASSIGNMENT 8

STATEMENT: *Run linear regressions and to interpret the output Using the preexisting data file.*

THEORY:

1. *Reading Data:*

- *The code begins by loading data from a CSV file that the user selects interactively. This data likely contains information about selling prices and kilometers driven from a car resale company.*

2. *Exploring Data:*

- *The code performs initial data exploration:*
- *View(data) opens a viewer to visually inspect the loaded data.*
- *dim(data) finds and displays the number of rows and columns in the dataset.*
- *str(data) shows the structure of the data, including data types.*

3. *Creating a Contingency Table:*

- *This part of the code creates a table (contingency table) that summarizes the counts of different combinations of selling prices and kilometers driven.*
- *margin.table(tab1) likely calculates and displays marginal totals of the table.*

4. *Linear Regression:*

- *The code performs linear regression, which is a statistical technique to model the relationship between two variables:*
- *model <- lm(selling_price..in.thousands. ~ km_driven..in.thousands., data=data) creates a linear regression model.*
- *summary(model) provides detailed information about the model, including coefficients, significance, and goodness-of-fit statistics.*

5. *Creating a Scatter Plot with Regression Line:*

- *The code generates a scatter plot to visualize the data points:*
- *The x-axis represents kilometers driven, and the y-axis represents selling prices.*
- *plot(...) is used to create the scatter plot.*
- *abline(model, col = "blue") adds a blue line to the plot, which represents the linear regression model.*

6. *Creating a ggplot2 Scatter Plot with Regression Line:*

- *This part of the code uses the ggplot2 library to create a more sophisticated scatter plot:*
- *library(ggplot2) loads the ggplot2 package.*
- *ggplot(...) initializes a ggplot2 plot with aesthetics and data mapping.*
- *geom_point() adds data points to the plot.*
- *geom_smooth(method = "lm", se = FALSE) adds a smooth (regression) line to the plot.*
- *labs(...) sets labels and the plot's title.*

SOURCE CODE:

```
data<-read.csv(file.choose())
```

```
View(data)

dim(data)

str(data)

tab1 =table(data$ selling_price..in.thousands. ,data$km_driven..in.thousands.)

margin.table(tab1)

# Perform linear regression

model <- lm(selling_price..in.thousands. ~ km_driven..in.thousands.,
data=data)

summary(model)

plot(data$km_driven..in.thousands., data$selling_price..in.thousands.,

    xlab = "km driven (in thousands)",

    ylab = "selling price (in thousands)",

    main = "Linear Regression")

# Add the regression line to the plot

abline(model, col = "blue")

library(ggplot2)

data.graph <- ggplot(data, aes(x = km_driven..in.thousands., y =
selling_price..in.thousands.)) +

  geom_point() +

  geom_smooth(method = "lm", se = FALSE) +  # Add a regression line

  labs(x = "km driven (in thousands)", y = "selling price (in thousands)", title =
"Linear Regression")
```

OUTPUT:

```
[1] 24  8
> str(data)
'data.frame':   24 obs. of  8 variables:
 $ name                     : chr  "Maruti 800 AC" "Maruti Wagon R LXI Minor" "Hyundai Verna 1.6 SX" "Datsu
n RediGO T Option" ...
 $ year                     : int  2007 2007 2012 2017 2014 2007 2016 2014 2015 2017 ...
 $ selling_price..in.thousands.: int  60 135 600 250 450 140 550 240 850 365 ...
 $ km_driven..in.thousands.    : int  70 50 100 46 141 125 25 60 25 78 ...
 $ fuel                     : chr  "Petrol" "Petrol" "Diesel" "Petrol" ...
 $ seller_type              : chr  "Individual" "Individual" "Individual" "Individual" ...
 $ transmission             : chr  "Manual" "Manual" "Manual" "Manual" ...
 $ owner                    : chr  "First Owner" "First Owner" "First Owner" "First Owner" ...
> tab1 =table(data$ selling_price..in.thousands. ,data$km_driven..in.thousands.)
> margin.table(tab1)
[1] 24
> # Perform linear regression
> model <- lm(selling_price..in.thousands. ~ km_driven..in.thousands., data=data)
> summary(model)
```

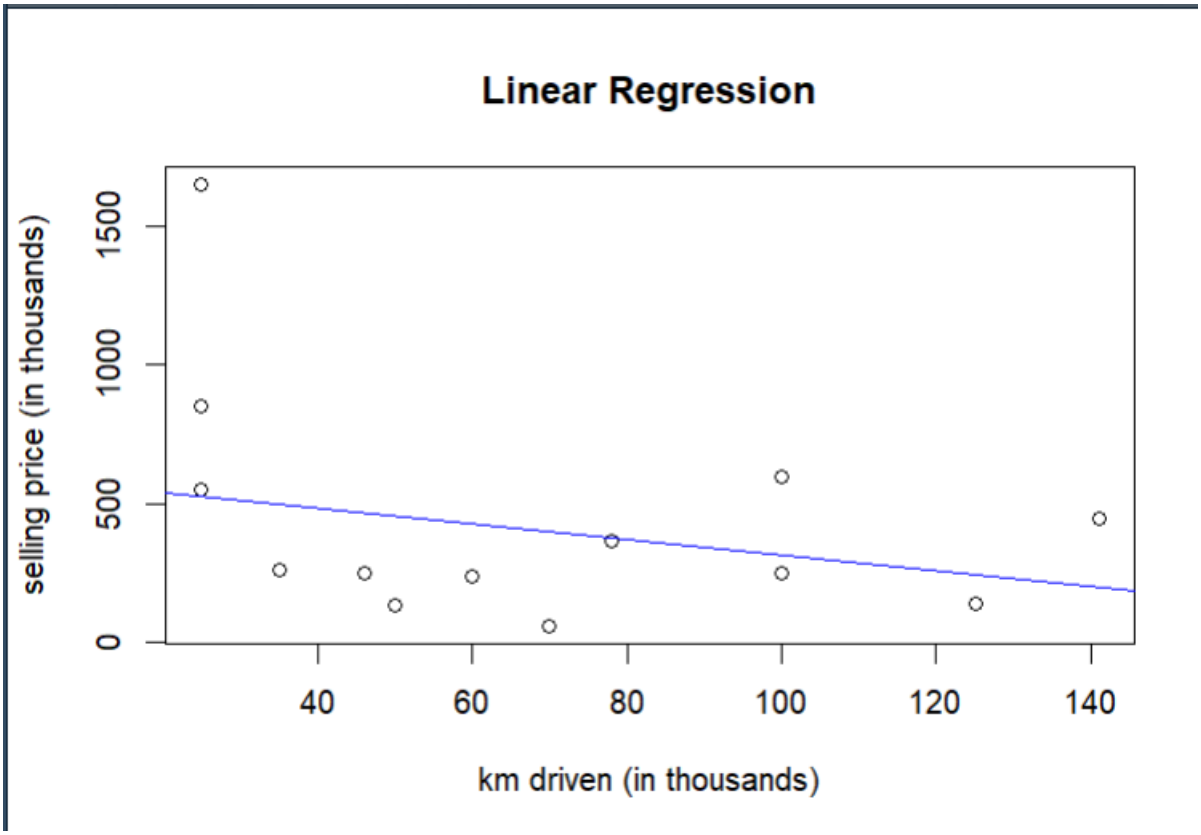
```
> model <- lm(selling_price..in.thousands. ~ km_driven..in.thousands., data=data)
> summary(model)

Call:
lm(formula = selling_price..in.thousands. ~ km_driven..in.thousands.,
    data = data)

Residuals:
    Min      1Q  Median      3Q     Max
-338.93 -221.12  -85.27  249.31 1125.42

Coefficients:
                      Estimate Std. Error t value Pr(>|t|)
(Intercept)            594.380    142.999   4.157 0.000412 ***
km_driven..in.thousands.   -2.792      1.840  -1.518 0.143352
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Residual standard error: 337.3 on 22 degrees of freedom
Multiple R-squared:  0.09477,   Adjusted R-squared:  0.05362
F-statistic: 2.303 on 1 and 22 DF,  p-value: 0.1434
```



CONCLUSION:

The provided R code serves as a comprehensive analysis and visualization tool for a dataset containing information on selling prices and kilometers driven from car resale company. It starts by loading and exploring the data, then constructs a contingency table to summarize the relationships between these variables. The primary focus of the code is on linear regression analysis, which helps uncover any linear associations between selling prices and kilometers driven.

The code generates both basic and enhanced scatter plots to visualize the data points and regression line, aiding in the interpretation of the linear relationship between the two variables. Overall, it provides valuable insights into how kilometers driven may impact selling prices, making it a useful tool for data analysis and decision-making in contexts where such relationships are important.