Practicals 9: Perform the Linear regression on the given datawarehouse data.

—---

```
# Create the data vectors.
x <- c(151, 174, 138, 186, 128, 136, 179, 163, 152, 131)
y <- c(63, 81, 56, 91, 47, 57, 76, 72, 62, 48)

# Apply the lm() function.
relation <- lm(y ~ x)

# Find the weight of a person with height 170.
a <- data.frame(x = 170)  # Corrected column name to match the linear regression model
result <- predict(relation, a)
print(result)

# Give the chart file a name.
png(file = "linearregression.png")

# Plot the chart.
plot(x, y, col = "blue", main = "Height & Weight Regression", abline(relation), cex = 1.3, pch =
16, xlab = "Height in cm", ylab = "Weight in Kg")

# Save the file.
dev.off()
```

OUTPUT:

```
       1
78.92713
```

plot(x, y, col = "blue", main = "Height & Weight Regression", abline(relation), cex = 1.3, pch = 16, xlab = "Height in cm", ylab = "Weight in Kg")

Where,

x, y: x and y are the data vectors to be plotted on the x-axis and y-axis, respectively.

col: col specifies the color of the points in the plot. In this case, the points will be displayed in blue.

main: main specifies the main title of the plot. The title is set to "Height & Weight Regression" in this example.

abline(relation): abline() is used to add a straight line to the plot. In this case, the relation object, which is the linear regression model (lm object), is passed to abline() to plot the regression line.

cex: cex stands for character expansion. It is used to control the size of characters (symbols or points) in the plot. Here, cex = 1.3 increases the size of the points by 30%.

pch:pch specifies the type of symbol or point to be used in the plot. The value 16 corresponds to solid circles.

xlab, ylab: xlab and ylab specify the labels for the x-axis and y-axis, respectively. In this example, "Height in cm" and "Weight in Kg" are used as axis labels.

Practical 10: Perform the logistic regression on the given data warehouse data.

https://rdrr.io/snippets/

Solution:

Multiple linear regression:

It is a machine learning technique used to predict the value of a dependent variable (here it is Admitted) based on two or more independent variables(here hsc,ssc,cet and viva). In other words, it helps to understand how different variables affect the outcome of something. **For example,** it can be used to predict a person's salary based on their age, education, and experience. The model estimates the relationship between the variables and uses this to make predictions. It is widely used in various fields such as finance, economics, engineering, and social sciences.

Following is a R code to determine whether a student will be admitted based on their marks in HSC, CET, and Viva. The total number of students is 70. The marks in each subject are out of 100. The dependent variable, 'Admitted', should be binary with a value of 1 indicating 'yes' and 0 indicating 'no'.

Code:

```
#Get Marks of 3 subjects, viva and status of admission
hsc <-
c(55,26,69,98,98,88,76,99,62,56,87,99,84,25,68,65,46,58,82,93,34,65,57,79,65,34,51,32,15,23,
22,11,79,65,34,51,32,15,23,22,11,10,57,59,64,65,22,23,23,45,4,8,6,35,35,36,39,35,36,68,69,64
,35,34,34,56,26,35,34,35)
ssc <-
c(32,35,31,24,68,65,46,58,82,93,34,99,84,25,31,65,24,53,39,32,35,31,24,68,65,46,58,82,93,34,
65,57,79,65,34,51,32,15,23,22,11,10,57,59,64,65,22,23,23,45,4,5,6,56,65,65,35,34,26,69,68,98
,69,68,69,78,45,65,68,69)
cet <-
c(55,26,69,98,98,88,76,99,62,56,87,99,84,25,31,65,24,53,39,32,35,31,24,68,65,46,58,82,93,34,
65,57,79,65,34,51,32,15,23,22,11,10,57,59,64,65,22,23,23,45,3,8,2,59,68,68,54,58,69,65,69,69
,65,58,57,59,69,68,58,65)
viva <-
c(68,65,46,58,82,93,34,65,57,79,65,34,51,32,15,23,24,53,39,32,35,31,24,68,65,46,58,82,93,34,
65,57,79,65,34,51,32,15,23,22,34,51,32,15,23,22,11,10,57,59,5,8,6,78,98,54,69,68,98,58,69,58
,54,45,65,56,44,98,45,54)
```

```
admitted <-
c(0,0,0,0,1,1,0,1,1,1,0,0,1,0,0,0,0,1,1,0,0,0,0,1,1,0,1,0,0,0,0,0,1,1,0,1,0,0,0,0,0,0,0,0,0,0,0,0,0,1,
0,0,0,0,0,1,0,0,0,1,1,1,0,0,0,1,0,0,0,0)

# Create a data frame with the independent and dependent variables
data <- data.frame(hsc, ssc, cet, viva, admitted)

# Fit the multiple linear regression model
model <- lm(admitted ~ hsc + ssc + cet + viva, data)

# Print the model summary
summary(model)

# Predict the admission status for a new student
new_student <- data.frame(hsc = 67, ssc = 98, cet = 86, viva = 74)
predict(model, new_student)

# Plot the model
plot(model)
```

OUTPUT:

```
Call:
lm(formula = admitted ~ hsc + ssc + cet + viva, data = data)

Residuals:
     Min       1Q   Median       3Q      Max 
-0.77638 -0.22696 -0.05468  0.26600  0.84825 

Coefficients:
             Estimate Std. Error t value Pr(>|t|)    
(Intercept) -0.436456   0.113755  -3.837 0.000285 ***
hsc          0.011433   0.002208   5.177 2.36e-06 ***
ssc          0.010537   0.002615   4.030 0.000149 ***
cet         -0.013690   0.003550  -3.856 0.000267 ***
viva         0.007827   0.002363   3.313 0.001512 ** 
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 0.3395 on 65 degrees of freedom
Multiple R-squared:  0.4755,	Adjusted R-squared:  0.4432 
F-statistic: 14.73 on 4 and 65 DF,  p-value: 1.281e-08

        1 

  0.7640119
```