5.5.1 Simple Regression in R

Simple Linear Regression in R

For understanding the concept let's consider a salary dataset where it is given the value of the dependent variable(salary) for every independent variable(years experienced).

Salary dataset:

Years experienced	Salary
1.1	39343
1.3	46205
1.5	37731
2	43525
2.2	39891
2.9	56642
3	60150
3.2	54445
3.2	64445
3.7	57189

For general purposes, we define:

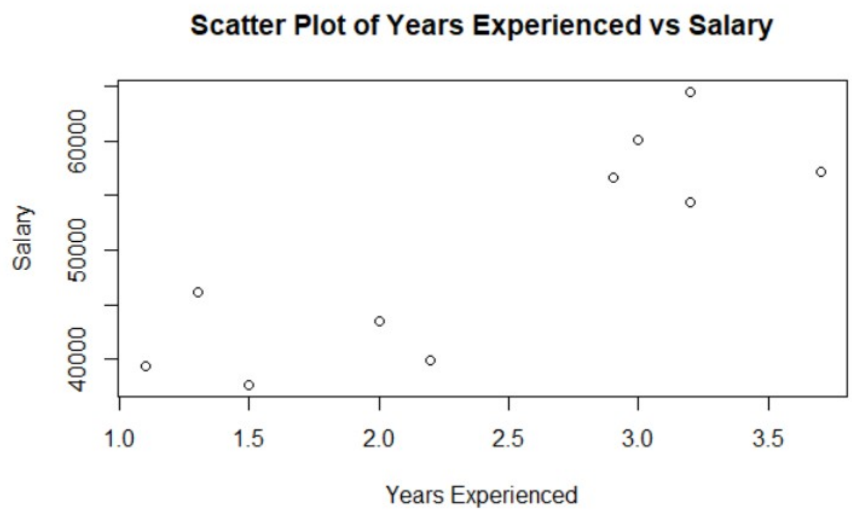
- x as a feature vector, i.e x = [x_1, x_2, …., x_n],
- y as a response vector, i.e y = [y_1, y_2, …., y_n]
- for n observations (in the above example, n=10).

First we convert these data values into R Data Frame

```
# Create the data frame
data <- data.frame(
  Years_Exp = c(1.1, 1.3, 1.5, 2.0, 2.2, 2.9, 3.0, 3.2, 3.2, 3.7),
  Salary = c(39343.00, 46205.00, 37731.00, 43525.00,
             39891.00, 56642.00, 60150.00, 54445.00, 64445.00, 57189.00)
)
```

Scatter plot of the given dataset

```
# Create the scatter plot
plot(data$Years_Exp, data$Salary,
     xlab = "Years Experienced",
     ylab = "Salary",
     main = "Scatter Plot of Years Experienced vs Salary")
```



Now, we have to find a line that fits the above scatter plot through which we can predict any value of y or response for any value of x
The line which best fits is called the Regression line.

The equation of the regression line is given by:

y = a + bx
Where y is the predicted response value, a is the y-intercept, x is the feature value and b is the slope.
To create the model, let's evaluate the values of regression coefficients a and b. And as soon as the estimation of these coefficients is done, the response model can be predicted.

The basic syntax for regression analysis in R is

```
lm(Y ~ model)
```

where Y is the object containing the dependent variable to be predicted and the model is the formula for the chosen mathematical model.
The command lm( ) provides the model's coefficients but no further statistical information.

The following R code is used to implement Simple Linear Regression:

```
install.packages('caTools')
library(caTools)
split = sample.split(data$Salary, SplitRatio = 0.7)
trainingset = subset(data, split == TRUE)
testset = subset(data, split == FALSE)

# Fitting Simple Linear Regression to the Training set
lm.r= lm(formula = Salary ~ Years_Exp,
         data = trainingset)
#Summary of the model
summary(lm.r)
```

Output:

```
Call:
lm(formula = Salary ~ Years_Exp, data = trainingset)
Residuals:
      1       2       3       5       6       8      10
  463.1  5879.1 -4041.0 -6942.0  4748.0   381.9  -489.1
Coefficients:
            Estimate Std. Error t value Pr(>|t|)
(Intercept)    30927       4877   6.341  0.00144 **
Years_Exp       7230       1983   3.645  0.01482 *
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1
Residual standard error: 4944 on 5 degrees of freedom
Multiple R-squared:  0.7266,    Adjusted R-squared:  0.6719
F-statistic: 13.29 on 1 and 5 DF,  p-value: 0.01482
```

Call: Using the “lm” function, we will be performing a regression analysis of “Salary” against “Years_Exp” according to the formula displayed on this line.

- **Residuals**: Each residual in the “Residuals” section denotes the difference between the actual salaries and predicted values. These values are unique to each observation in the data set. For instance, observation 1 has a residual of 463.1.
- **Coefficients**: Linear regression coefficients are revealed within the contents of this section.
- **(Intercept)**: The estimated salary when Years_Exp is zero is 30927, which represents the intercept for this case.
- **Years_Exp**: For every year of experience gained, the expected salary is estimated to increase by 7230 units according to the coefficient for “Years_Exp”. This coefficient value suggests that each year of experience has a significant impact on the estimated salary.
- **Estimate**:The model’s estimated coefficients can be found in this column.
- **Std. Error**: “More precise estimates” can be deduced from smaller standard errors that are a gauge of the ambiguity that comes along with coefficient estimates.
- **t value**: The coefficient estimate’s standard error distance from zero is measured by the t-value. Its purpose is to examine the likelihood of the coefficient being zero by testing the null hypothesis. A higher t-value’s absolute value indicates a higher possibility of statistical significance pertaining to the coefficient.
- **Pr(>|t|)**: This column provides the p-value associated with the t-value. The p-value indicates the probability of observing the t-statistic (or more extreme) under the null hypothesis that the coefficient is zero. In this case, the p-value for the intercept is 0.00144, and for “Years_Exp,” it is 0.01482.
- **Signif. codes**: These codes indicate the level of significance of the coefficients.
- **Residual standard error**: This is a measure of the variability of the residuals. In this case, it’s

4944, which represents the typical difference between the actual salaries and the predicted salaries.

- **Multiple R-squared**: R-squared (R^2) is a measure of the goodness of fit of the model. It represents the proportion of the variance in the dependent variable that is explained by the independent variable(s). In this case, the R-squared is 0.7266, which means that approximately 72.66% of the variation in salaries can be explained by years of experience.
- **Adjusted R-squared**: The adjusted R-squared adjusts the R-squared value based on the number of predictors in the model. It accounts for the complexity of the model. In this case, the adjusted R-squared is 0.6719.
- **F-statistic**: The F-statistic is used to test the overall significance of the model. In this case, the F-statistic is 13.29 with 1 and 5 degrees of freedom, and the associated p-value is 0.01482. This p-value suggests that the model as a whole is statistically significant.

In summary, this linear regression analysis suggests that there is a significant relationship between years of experience (Years_Exp) and salary (Salary). The model explains approximately 72.66% of the variance in salaries, and both the intercept and the coefficient for “Years_Exp” are statistically significant at the 0.01 and 0.05 significance levels, respectively.

Predict values using predict function

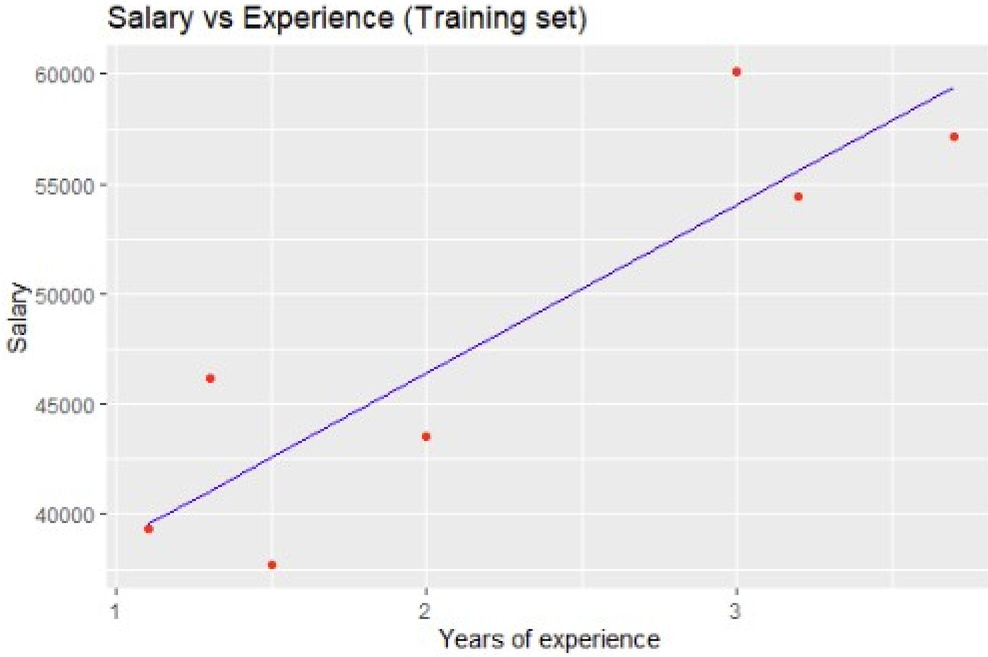
```
# Create a data frame with new input values
new_data <- data.frame(Years_Exp = c(4.0, 4.5, 5.0))

# Predict using the linear regression model
predicted_salaries <- predict(lm.r, newdata = new_data)

# Display the predicted salaries
print(predicted_salaries)
```

Output:

```
       1        2        3
65673.14 70227.40 74781.66
```

Visualizing the Training set results:

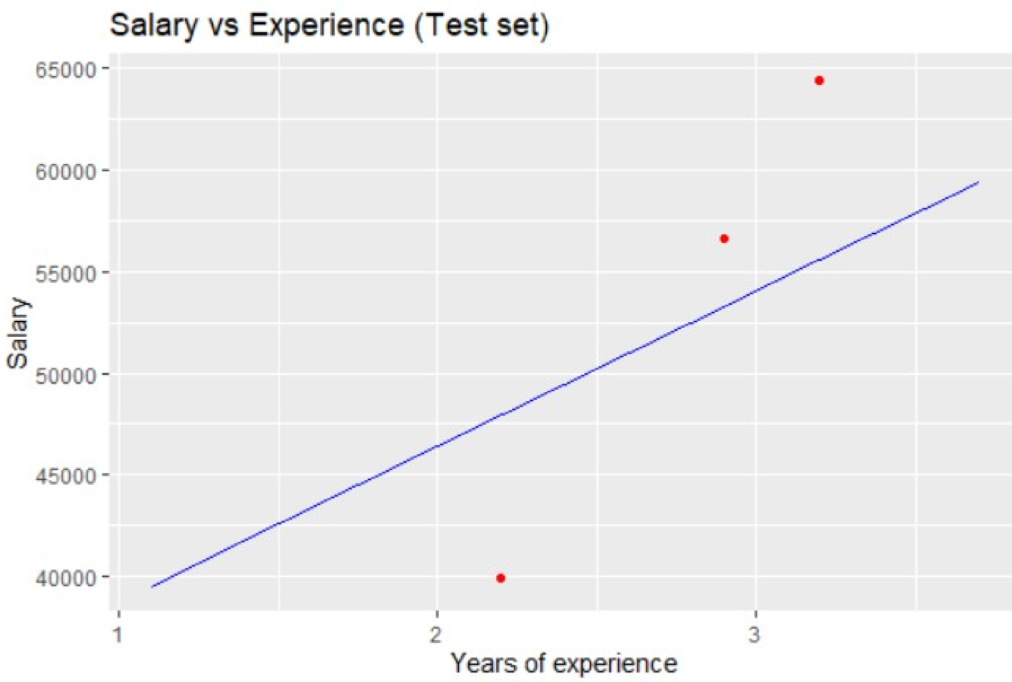
```
# Visualising the Training set results
ggplot() + geom_point(aes(x = trainingset$Years_Ex,
                          y = trainingset$Salary), colour = 'red') +
  geom_line(aes(x = trainingset$Years_Ex,
                y = predict(lm.r, newdata = trainingset)), colour = 'blue') +

  ggtitle('Salary vs Experience (Training set)') +
  xlab('Years of experience') +
  ylab('Salary')
```



Training data visualization in Linear Regression in R

Visualizing the Testing set results:

```
# Visualising the Test set results
ggplot() +
  geom_point(aes(x = testset$Years_Exp, y = testset$Salary),
             colour = 'red') +
  geom_line(aes(x = trainingset$Years_Exp,
                y = predict(lm.r, newdata = trainingset)),
            colour = 'blue') +
  ggtitle('Salary vs Experience (Test set)') +
  xlab('Years of experience') +
  ylab('Salary')
```



Testing Data Visualization in Linear Regression in R

Advantages of Simple Linear Regression in R:

1. Easy to implement: R provides built-in functions, such as lm(), to perform Simple Linear Regression quickly and efficiently.
2. Easy to interpret: Simple Linear Regression models are easy to interpret, as they model a linear relationship between two variables.
3. Useful for prediction: Simple Linear Regression can be used to make predictions about the dependent variable based on the independent variable.
4. Provides a measure of goodness of fit: Simple Linear Regression provides a measure of how well the model fits the data, such as the R-squared value.

Disadvantages of Simple Linear Regression in R:

1. Assumes linear relationship: Simple Linear Regression assumes a linear relationship between the variables, which may not be true in all cases.
2. Sensitive to outliers: Simple Linear Regression is sensitive to outliers, which can significantly affect the model coefficients and predictions.
3. Assumes independence of observations: Simple Linear Regression assumes that the observations are independent, which may not be true in some cases, such as time series data.
4. Cannot handle non-numeric data: Simple Linear Regression can only handle numeric data and cannot be used for categorical or non-numeric data.
5. Overall, Simple Linear Regression is a useful tool for modeling the relationship between two variables, but it has some limitations and assumptions that need to be carefully considered.