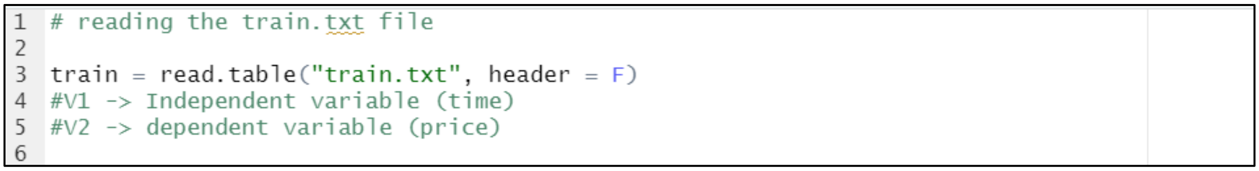
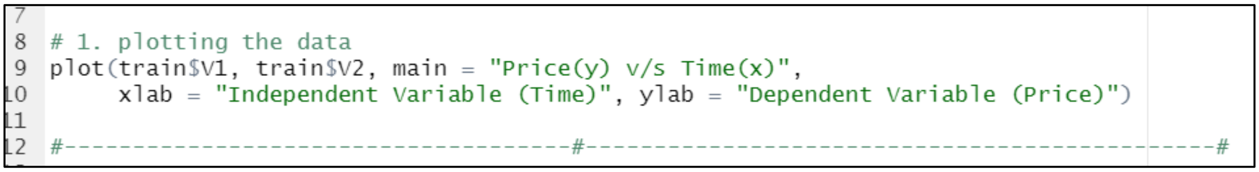
Code for reading the train data file



```
# reading the train.txt file

train = read.table("train.txt", header = F)
#V1 -> Independent variable (time)
#V2 -> dependent variable (price)

```

Answer 1

Code for plotting the data



```

# 1. plotting the data
plot(train$V1, train$V2, main = "Price(y) v/s Time(x)",
     xlab = "Independent Variable (Time)", ylab = "Dependent Variable (Price)")

#--------------------------------------#---------------------------------------------#
```

Plot of the Data showing a linear relationship between the two variables.

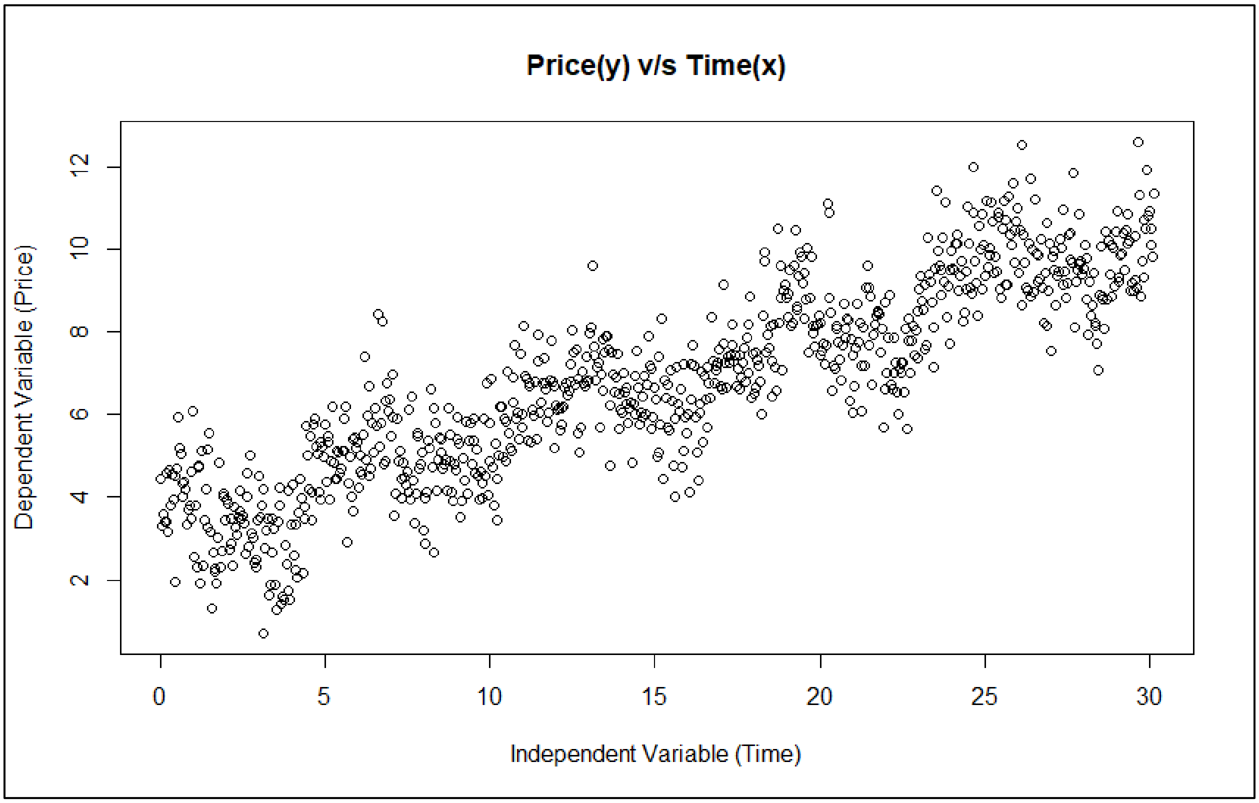
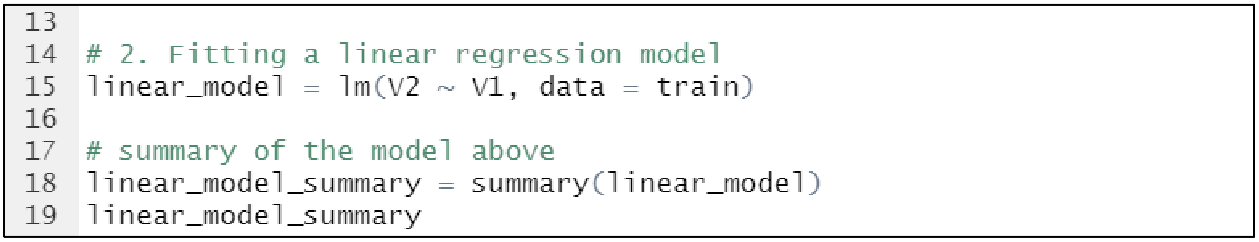


Figure 1.1

##--##--##

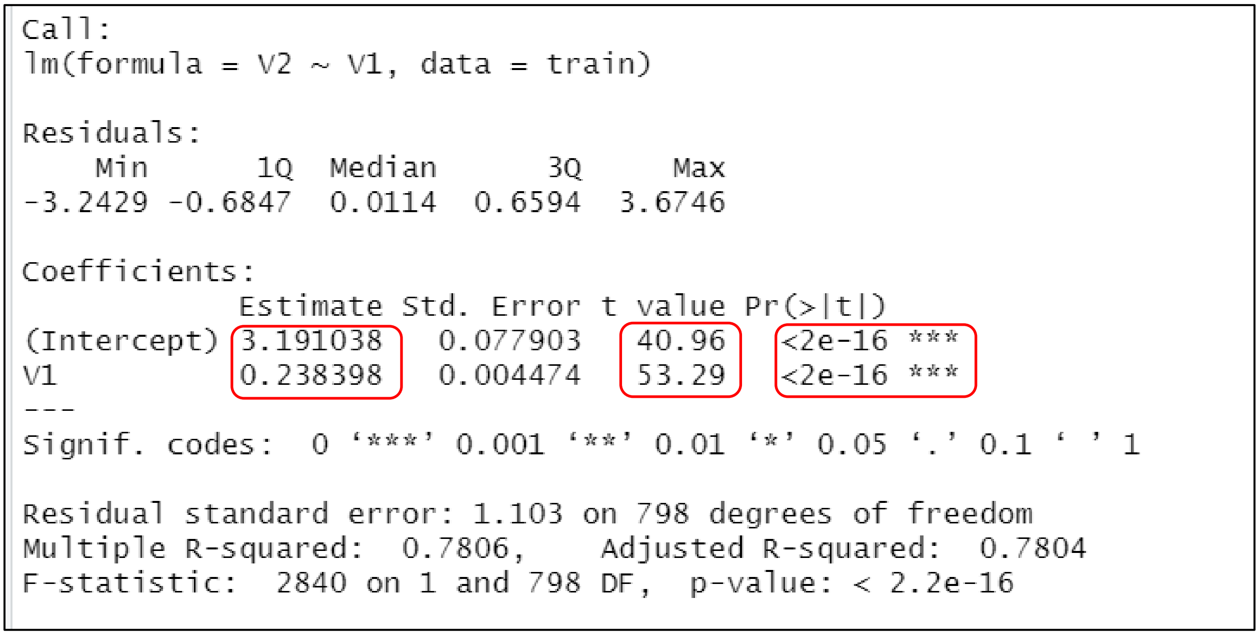
Answer 2

Code to fit the model and display its summary



```

# 2. Fitting a linear regression model
linear_model = lm(V2 ~ V1, data = train)

# summary of the model above
linear_model_summary = summary(linear_model)
linear_model_summary
```

Output



```
Call:
lm(formula = V2 ~ V1, data = train)

Residuals:
    Min      1Q  Median      3Q     Max 
-3.2429 -0.6847  0.0114  0.6594  3.6746 

Coefficients:
            Estimate Std. Error t value Pr(>|t|)    
(Intercept) 3.191038   0.077903   40.96   <2e-16 ***
V1          0.238398   0.004474   53.29   <2e-16 ***
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Residual standard error: 1.103 on 798 degrees of freedom
Multiple R-squared:  0.7806,	Adjusted R-squared:  0.7804 
F-statistic:  2840 on 1 and 798 DF,  p-value: < 2.2e-16
```

Ans 2a.

From the output above (highlighted in red),

$$\boldsymbol{\hat{\beta}_0 = 3.191038}$$

$$\boldsymbol{\hat{\beta}_1 = 0.238398}$$

Ans 2b.

To find out the significance of $\boldsymbol{\beta_1}$, we conduct a two-tailed hypothesis test as follows –

$$\mathbf{H0}: \boldsymbol{\beta_1 = 0}$$

$\mathbf{H1}: \boldsymbol{\beta_1} \neq \mathbf{0}$

From the model summary output (highlighted in red), we can see that the t-statistic for this hypothesis test if **53.29**, and the corresponding p-value is **< 2x10^-16**, thus, we can reject the Null Hypothesis even at 0.1% level of significance.

Therefore, there is sufficient evidence to conclude that $\boldsymbol{\beta_1}$ is statistically significantly different from 0, i.e., there is a linear trend present in the data.

Ans 2c.

To find out the significance of $\boldsymbol{\beta_0,}$ we conduct a two-tailed hypothesis test as follows –

$\mathbf{H0}: \boldsymbol{\beta_0} = \mathbf{0}$

$\mathbf{H1}: \boldsymbol{\beta_0} \neq \mathbf{0}$

From the model summary output (highlighted in red), we can see that the t-statistic for this hypothesis test if **40.96**, and the corresponding p-value is **< 2x10^-16**, thus, we can reject the Null Hypothesis even at 0.1% level of significance.

Therefore, there is sufficient evidence to conclude that $\boldsymbol{\beta_0}$ is statistically significantly different from 0.

##--##--##

Answer 3

Procedure used

Step 1: a regression model was estimated using the linear independent variable and all the non-linear forms of the independent variables as mentioned in the question. So, the first regression model was as follows –

$$V2 = \boldsymbol{\beta_0} + \boldsymbol{\beta_1}V1 + \boldsymbol{\beta_2 cos(V1)} + \boldsymbol{\beta_3 log(V1)} + \boldsymbol{\beta_4 cos(4V1)} + \boldsymbol{\beta_5 sin(3V1)} + \boldsymbol{\beta_6 sin(5V1)} + \boldsymbol{\beta_7 sin(2V1)} * \boldsymbol{cos(2V1)}$$

Step 2: Remove the least significant variable from the above model (the one with the lowest value of t-statistic or the highest p-value).

Note that we only remove the variable if it is not significant at 1% confidence level.

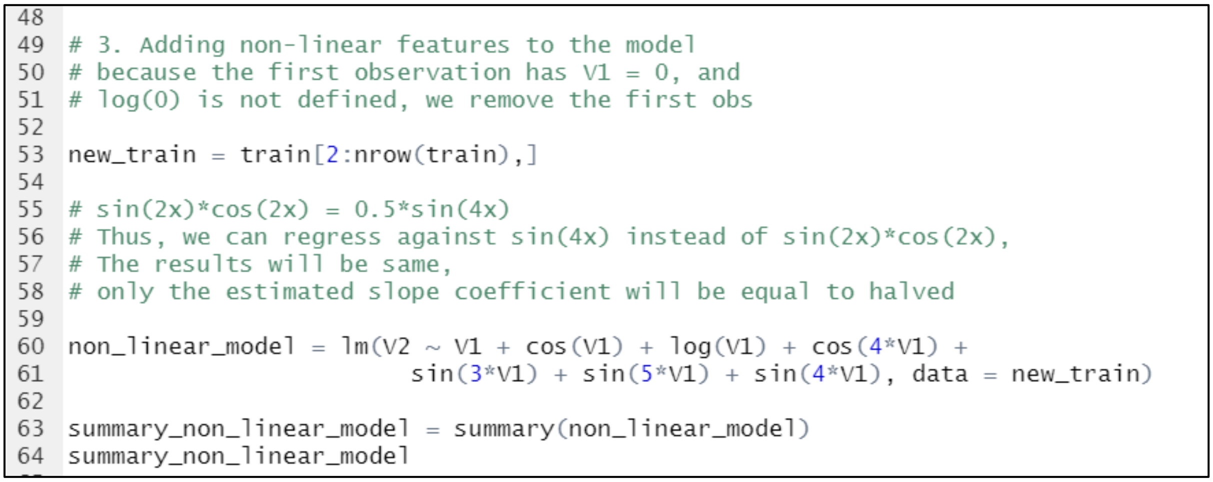
Step 3: Check if the F-statistic of the model improves after above elimination.

- If yes, then go to Step 2 again (i.e, remove the next least significant variable),
- otherwise, the initial model is the best model.

Repeat the above process until the F-statistic of the model can no longer be improved by removing the least significant variable.

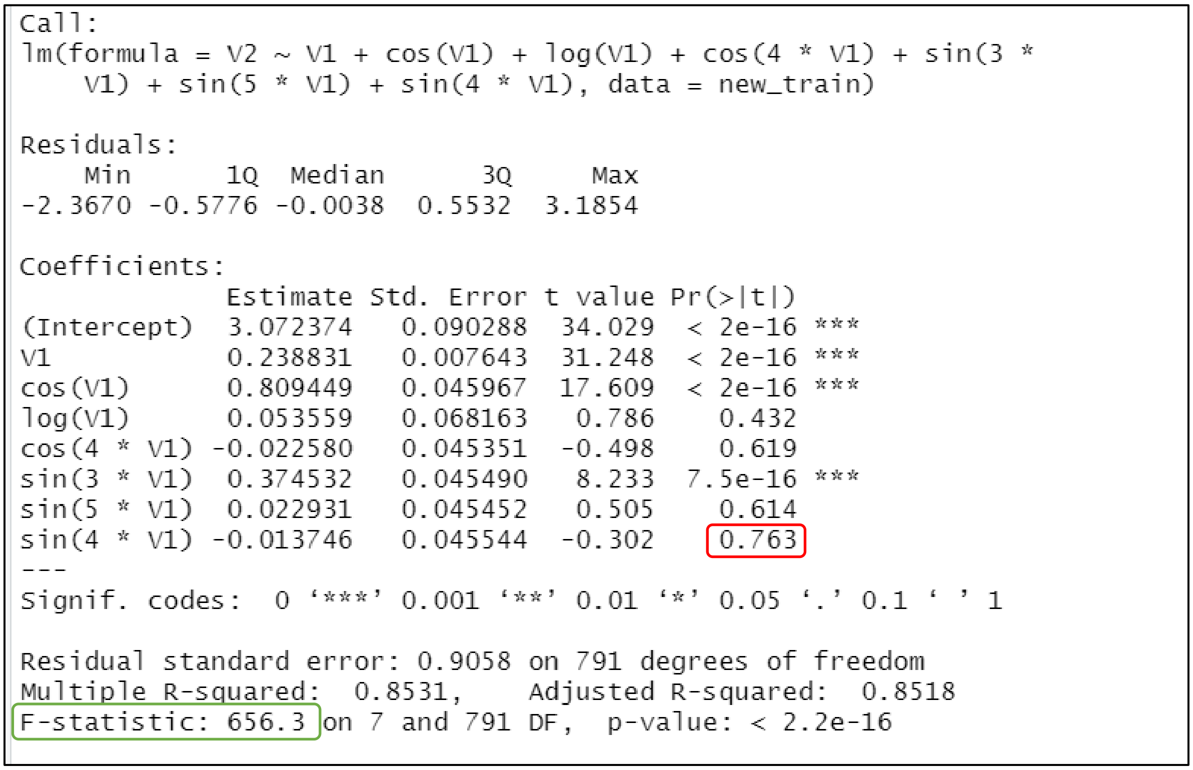
This procedure will remove the least significant variables one by one, but only if removal of that variable results in the improvement of overall model significance (as indicated by the model F-statistic). The final model obtained will contain only those variables which help improve the overall model significance, and thus, it can be considered the best model.

Code for the first model involving all the variables



```
# 3. Adding non-linear features to the model
# because the first observation has V1 = 0, and
# log(0) is not defined, we remove the first obs

new_train = train[2:nrow(train),]

# sin(2x)*cos(2x) = 0.5*sin(4x)
# Thus, we can regress against sin(4x) instead of sin(2x)*cos(2x),
# The results will be same,
# only the estimated slope coefficient will be equal to halved

non_linear_model = lm(V2 ~ V1 + cos(V1) + log(V1) + cos(4*V1) +
                        sin(3*V1) + sin(5*V1) + sin(4*V1), data = new_train)

summary_non_linear_model = summary(non_linear_model)
summary_non_linear_model
```

Output



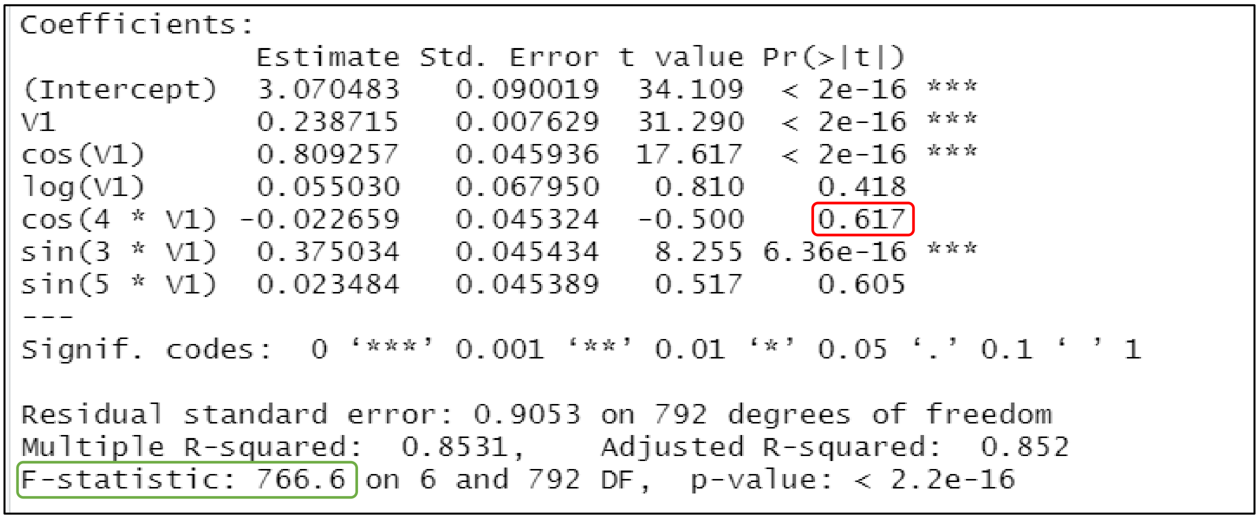
```
Call:
lm(formula = V2 ~ V1 + cos(V1) + log(V1) + cos(4 * V1) + sin(3 *
    V1) + sin(5 * V1) + sin(4 * V1), data = new_train)

Residuals:
    Min      1Q  Median      3Q     Max
-2.3670 -0.5776 -0.0038  0.5532  3.1854

Coefficients:
             Estimate Std. Error t value Pr(>|t|)
(Intercept)  3.072374   0.090288  34.029  < 2e-16 ***
V1           0.238831   0.007643  31.248  < 2e-16 ***
cos(V1)      0.809449   0.045967  17.609  < 2e-16 ***
log(V1)      0.053559   0.068163   0.786    0.432
cos(4 * V1) -0.022580   0.045351  -0.498    0.619
sin(3 * V1)  0.374532   0.045490   8.233  7.5e-16 ***
sin(5 * V1)  0.022931   0.045452   0.505    0.614
sin(4 * V1) -0.013746   0.045544  -0.302    0.763
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Residual standard error: 0.9058 on 791 degrees of freedom
Multiple R-squared:  0.8531,    Adjusted R-squared:  0.8518
F-statistic: 656.3 on 7 and 791 DF,  p-value: < 2.2e-16
```

Table 3.1

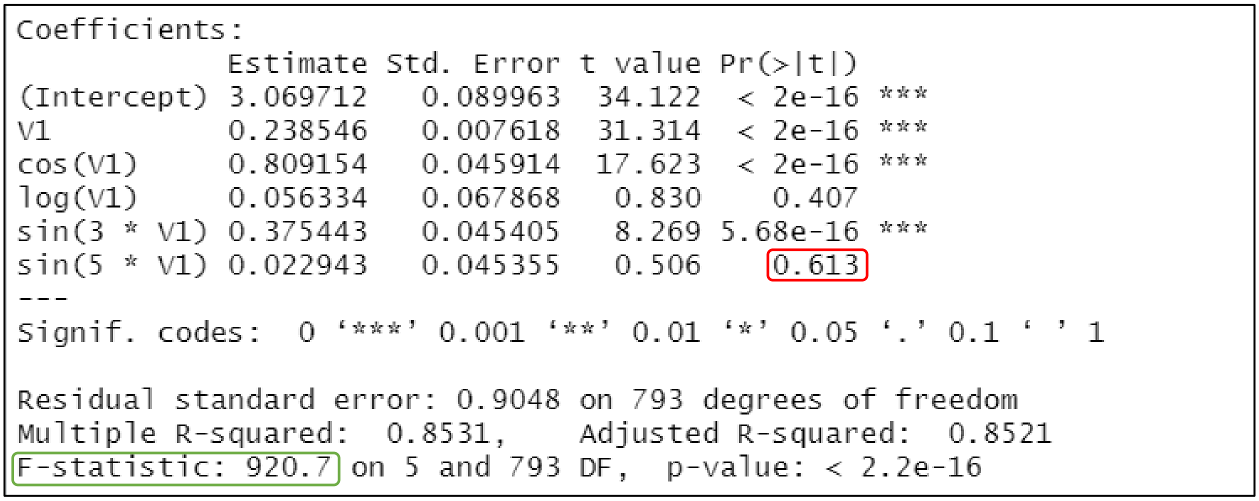
As we can see from the summary above (Table 3.1), even though the overall model is significant (indicated by a high F-statistic), there are many variables whose slope estimates are not statistically significantly different from 0 even at 10% significance level as indicated by their respective high p-values (e.g., log(V1), cos(4V1), sin(5V1) etc.). Thus, there might be scope to further improve this model.

So, we remove the variable which is the least significant, i.e., the one with the highest p-value: **sin(4V1)** and repeat this process until either F-statistic is improving or all the remaining variables are significant at 1% significance level.

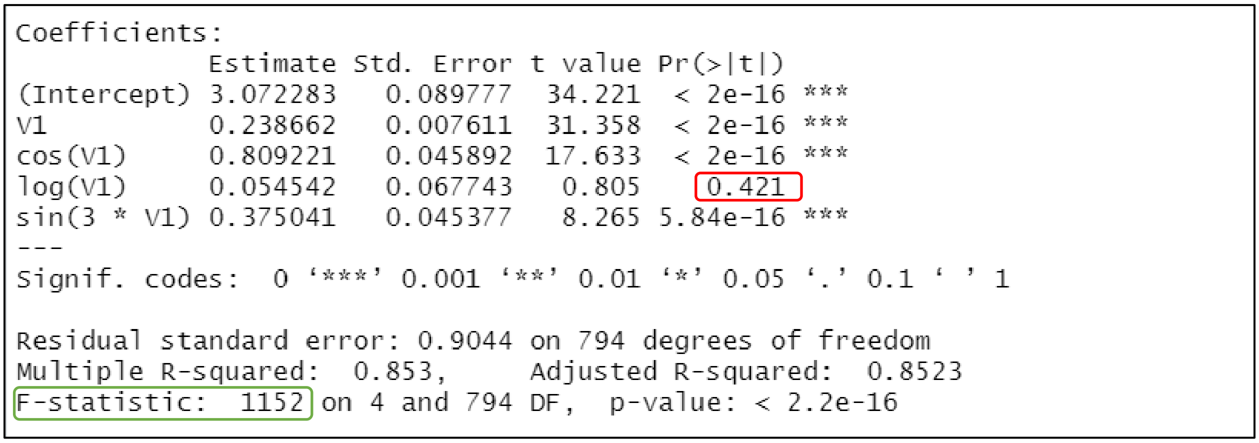
Outputs for intermediate models (From top to bottom: Tables 3.2, 3.3, and 3.4 respectively)



```
Coefficients:
             Estimate Std. Error t value Pr(>|t|)
(Intercept)  3.070483   0.090019  34.109  < 2e-16 ***
V1           0.238715   0.007629  31.290  < 2e-16 ***
cos(V1)      0.809257   0.045936  17.617  < 2e-16 ***
log(V1)      0.055030   0.067950   0.810    0.418
cos(4 * V1) -0.022659   0.045324  -0.500    0.617
sin(3 * V1)  0.375034   0.045434   8.255 6.36e-16 ***
sin(5 * V1)  0.023484   0.045389   0.517    0.605
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Residual standard error: 0.9053 on 792 degrees of freedom
Multiple R-squared:  0.8531,	Adjusted R-squared:  0.852
F-statistic: 766.6 on 6 and 792 DF,  p-value: < 2.2e-16
```

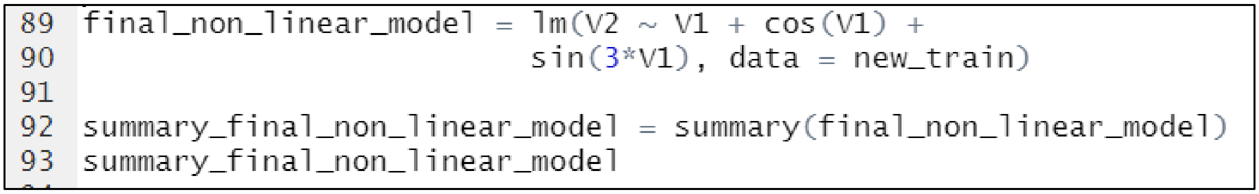


```
Coefficients:
            Estimate Std. Error t value Pr(>|t|)
(Intercept) 3.069712   0.089963  34.122  < 2e-16 ***
V1          0.238546   0.007618  31.314  < 2e-16 ***
cos(V1)     0.809154   0.045914  17.623  < 2e-16 ***
log(V1)     0.056334   0.067868   0.830    0.407
sin(3 * V1) 0.375443   0.045405   8.269 5.68e-16 ***
sin(5 * V1) 0.022943   0.045355   0.506    0.613
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Residual standard error: 0.9048 on 793 degrees of freedom
Multiple R-squared:  0.8531,	Adjusted R-squared:  0.8521
F-statistic: 920.7 on 5 and 793 DF,  p-value: < 2.2e-16
```



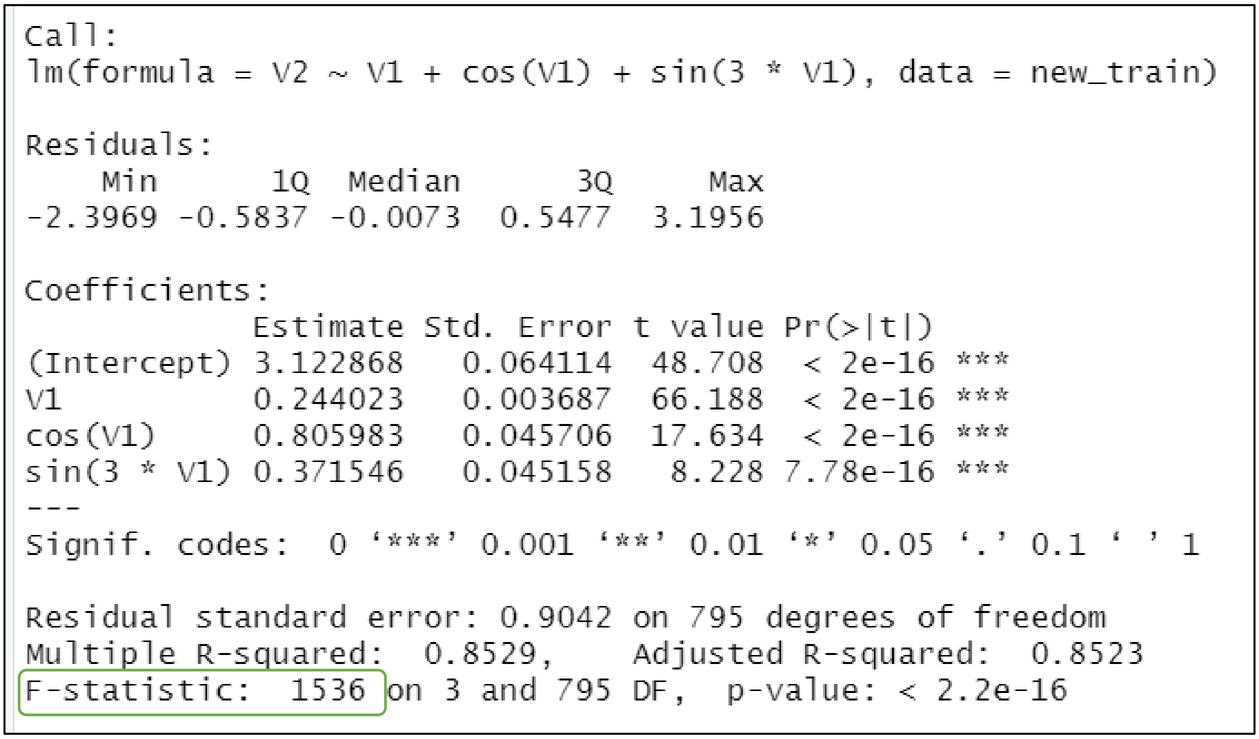
```
Coefficients:
            Estimate Std. Error t value Pr(>|t|)
(Intercept) 3.072283   0.089777  34.221  < 2e-16 ***
V1          0.238662   0.007611  31.358  < 2e-16 ***
cos(V1)     0.809221   0.045892  17.633  < 2e-16 ***
log(V1)     0.054542   0.067743   0.805    0.421
sin(3 * V1) 0.375041   0.045377   8.265 5.84e-16 ***
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Residual standard error: 0.9044 on 794 degrees of freedom
Multiple R-squared:  0.853,	Adjusted R-squared:  0.8523
F-statistic:  1152 on 4 and 794 DF,  p-value: < 2.2e-16
```

Code for Final Model



```
final_non_linear_model = lm(V2 ~ V1 + cos(V1) +
                           sin(3*V1), data = new_train)

summary_final_non_linear_model = summary(final_non_linear_model)
summary_final_non_linear_model
```

Final Model Output



```
Call:
lm(formula = V2 ~ V1 + cos(V1) + sin(3 * V1), data = new_train)

Residuals:
    Min      1Q  Median      3Q     Max 
-2.3969 -0.5837 -0.0073  0.5477  3.1956 

Coefficients:
            Estimate Std. Error t value Pr(>|t|)    
(Intercept) 3.122868   0.064114  48.708  < 2e-16 ***
V1          0.244023   0.003687  66.188  < 2e-16 ***
cos(V1)     0.805983   0.045706  17.634  < 2e-16 ***
sin(3 * V1) 0.371546   0.045158   8.228 7.78e-16 ***
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Residual standard error: 0.9042 on 795 degrees of freedom
Multiple R-squared:  0.8529,	Adjusted R-squared:  0.8523 
F-statistic:  1536 on 3 and 795 DF,  p-value: < 2.2e-16
```

Table 3.5

After removing 4 out of 7 variables considered initially,

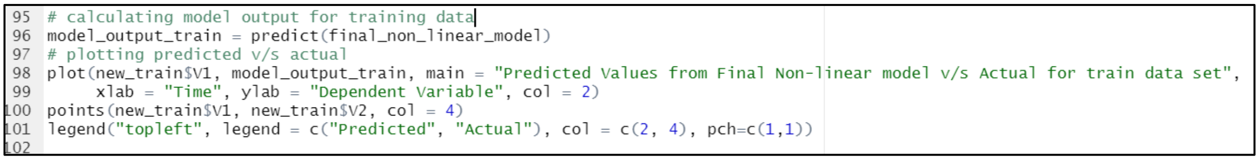
- F-statistic increased from 656 to 1536 – indicating improvement in overall model significance
- Adjusted R-squared increased from 85.18% to 85.23% - indicating the final model explains slightly higher variance in V2 than the initial model (after adjusting for model complexity)
- In the final model, all the variables are significant at even 0.1% significance level.

Thus, the “best” model includes the variables V1, cos(V1), and sin(3V1), and the model equation is –

$$\boldsymbol{V2 = 3.1229 + 0.2440V1 + 0.8060cos(V1) + 0.3715sin(3V1)}$$

Plotting the model v/s actual output for training dataset

Code



```
# calculating model output for training data
model_output_train = predict(final_non_linear_model)
# plotting predicted v/s actual
plot(new_train$V1, model_output_train, main = "Predicted Values from Final Non-linear model v/s Actual for train data set",
     xlab = "Time", ylab = "Dependent Variable", col = 2)
points(new_train$V1, new_train$V2, col = 4)
legend("topleft", legend = c("Predicted", "Actual"), col = c(2, 4), pch=c(1,1))
```

Output

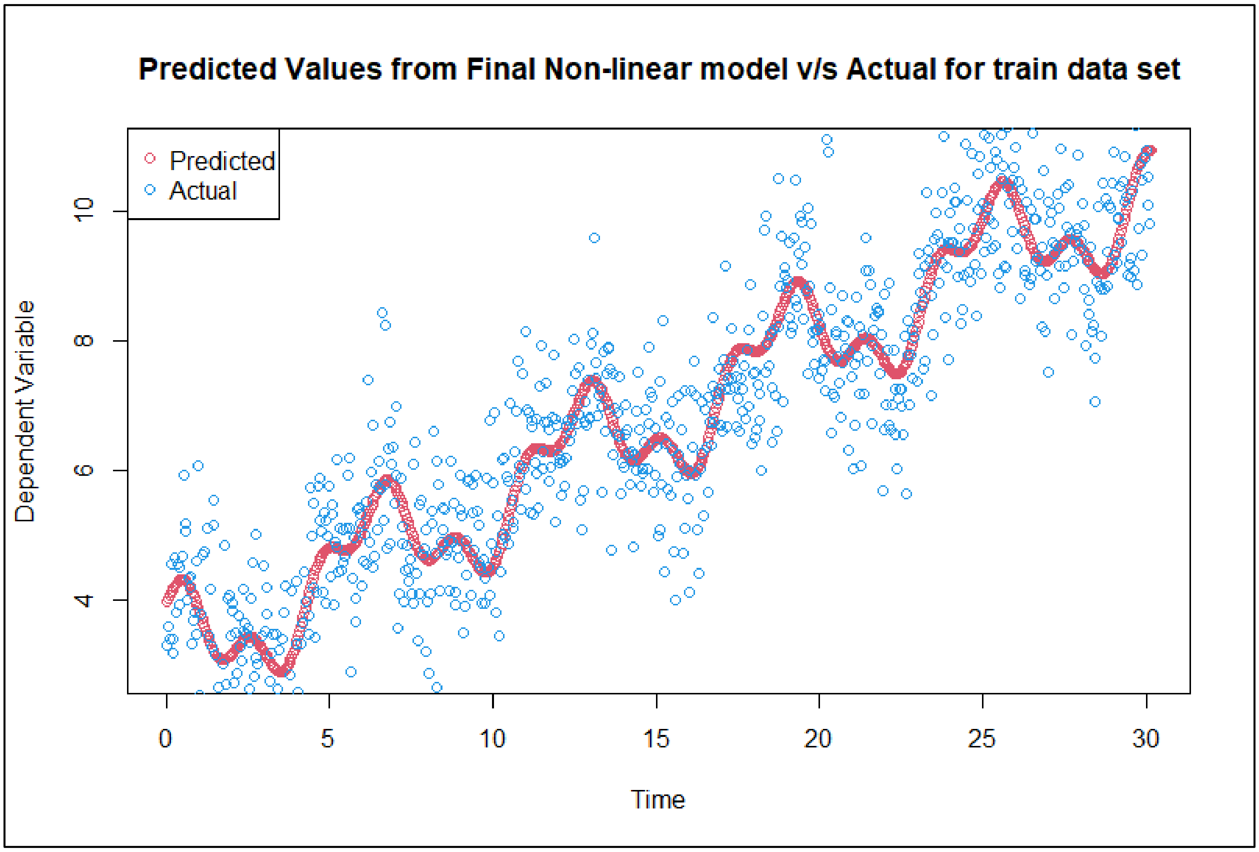


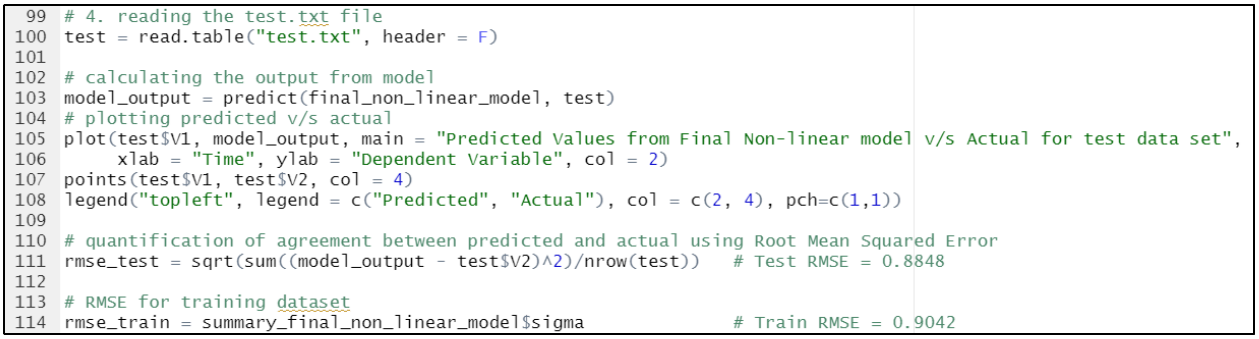
Figure 3.1

From the plot above, we can see that the model captures the non-linear trend in the training data quite well. Thus, this model can be concluded as the “best” both quantitively and visually.

##--##--##

Answer 4

Code



```
# 4. reading the test.txt file
test = read.table("test.txt", header = F)

# calculating the output from model
model_output = predict(final_non_linear_model, test)
# plotting predicted v/s actual
plot(test$V1, model_output, main = "Predicted Values from Final Non-linear model v/s Actual for test data set",
     xlab = "Time", ylab = "Dependent Variable", col = 2)
points(test$V1, test$V2, col = 4)
legend("topleft", legend = c("Predicted", "Actual"), col = c(2, 4), pch=c(1,1))

# quantification of agreement between predicted and actual using Root Mean Squared Error
rmse_test = sqrt(sum((model_output - test$V2)^2)/nrow(test))    # Test RMSE = 0.8848

# RMSE for training dataset
rmse_train = summary_final_non_linear_model$sigma              # Train RMSE = 0.9042
```

Output

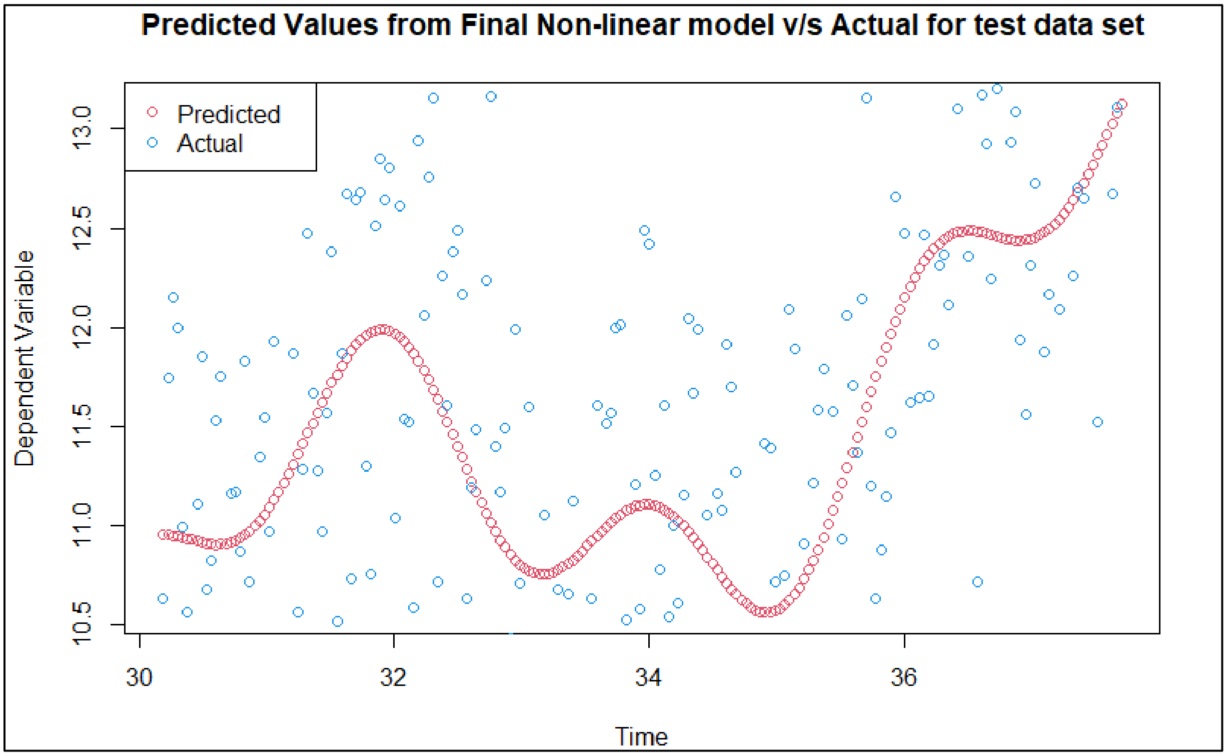


Figure 4.1

As evidenced by the plot above, the predicted and the actual test data does not agree very well, but the model does seem to pass through the average of the test data points, i.e., it has similar no. of points above and below it. This may be credited to a poorly defined relationship between the two variables in the test data set as compared to the training dataset. Also, because we used a non-linear model, this might also be an overfitting issues – i.e., the model is not generalizable.

In order to quantify the performance, we calculate the Root Mean Squared Error, which comes out to be 0.8848 for the test dataset, and 0.9042 for training dataset. This indicates that the model predictions are in fact in line with the test data.