Regression

Aloksai Choudari, Alekhya Pinnamaneni

2022-09-25

How does linear regression work?

Linear regression essentially aims to create an accurate model of the relationship between two variables based on provided data. Linear regression fits a straight line to the data while trying to minimize the gap between the line and the actual data points, or the residual sum of squares (RSS). The lower this value is, the more accurate the linear model is. Linear regression makes it straightforward and simple to create and understand a model for any set of data. It also prevents overfitting, which is a problem that occurs when a model is so closely fit to a data set that it becomes unusable for other data sets. However, linear regression can also be too simple for some data sets. If a data set has one outlier, it can completely throw off the linear model making it much less accurate. Linear regression can be very useful for some data sets, but it is important to evaluate a data set to determine which type of model will be the most accurate.

Select a data set

Data set: Metro Interstate Traffic Volume Data

Source: https://archive.ics.uci.edu/ml/datasets/Metro+Interstate+Traffic+Volume

Target column: 'traffic_volume'

No. of rows: 48,205 rows

Load the data

```
traffic <- read.csv("traffic.csv")
```

Split the data into train and test data

```
set.seed(1234)
sample <- sample(1:nrow(traffic), nrow(traffic)*0.8, replace=FALSE)
train <- traffic[sample,]
test <- traffic[-sample,]
```

Data exploration on the train data

```
attach(train)

# Prints the first 10 rows of the train data for traffic
head(train, n=10)
```

```
##       holiday   temp rain_1h snow_1h clouds_all weather_main
## 40784    None 274.29    0.00       0         90       Clouds
## 40854    None 273.68    0.00       0         90         Snow
## 41964    None 273.51    0.00       0          5        Clear
## 15241    None 292.74    0.00       0          1        Clear
## 33702    None 290.08    0.00       0         90         Mist
## 35716    None 293.65    0.00       0          1        Clear
## 17487    None 290.43    3.74       0         90         Mist
## 15220    None 286.59    0.00       0          0        Clear
## 19838    None 277.28    0.00       0         90       Clouds
## 2622     None 264.20    0.00       0         90         Snow
##       weather_description       date_time traffic_volume
## 40784     overcast clouds 1/19/2018 11:00           4936
## 40854          light snow  1/22/2018 5:00           2797
## 41964        sky is clear  3/1/2018 20:00           3169
## 15241        sky is clear   7/5/2014 4:00            319
## 33702                mist  5/17/2017 3:00            341
## 35716        sky is clear 7/26/2017 23:00           3420
## 17487                mist 8/18/2015 12:00           4751
## 15220        Sky is Clear   7/4/2014 4:00            382
## 19838     overcast clouds 12/5/2015 18:00           4662
## 2622            heavy snow  1/3/2013 11:00           4280
```

```
# Prints the mean of the temperature in kelvin
mean(temp)
```

```
## [1] 281.1764
```

```
# Prints the most common weather condition
names(which.max(table(weather_main)))
```

```
## [1] "Clouds"
```

```
# Prints all the holidays
unique(holiday)
```

```
##  [1] "None"                      "New Years Day"
##  [3] "Independence Day"          "Christmas Day"
##  [5] "Washingtons Birthday"      "Memorial Day"
##  [7] "State Fair"                "Thanksgiving Day"
##  [9] "Labor Day"                 "Martin Luther King Jr Day"
## [11] "Columbus Day"              "Veterans Day"
```

```
# Prints smallest and largest amount of hourly rain in mm
range(rain_1h)
```

```
## [1]    0.0 9831.3
```

```
# Print the median percentage of cloud cover
median(clouds_all)
```

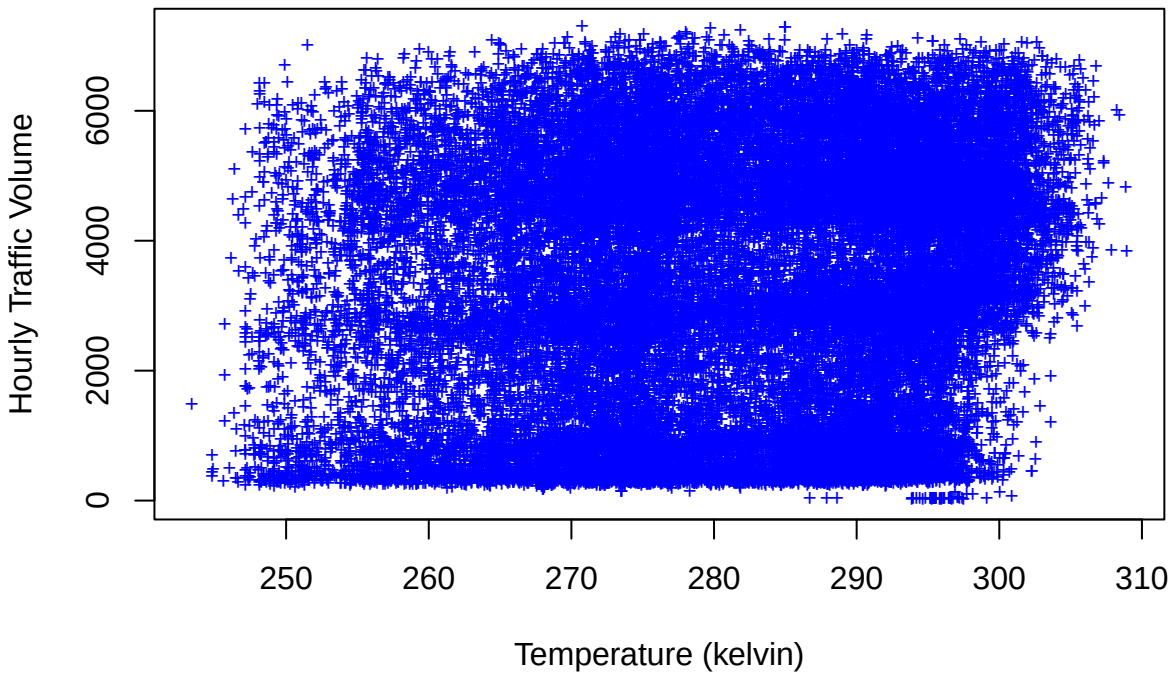
```
## [1] 64
```

```
# Prints statistics for hourly traffic volume
summary(traffic_volume)
```

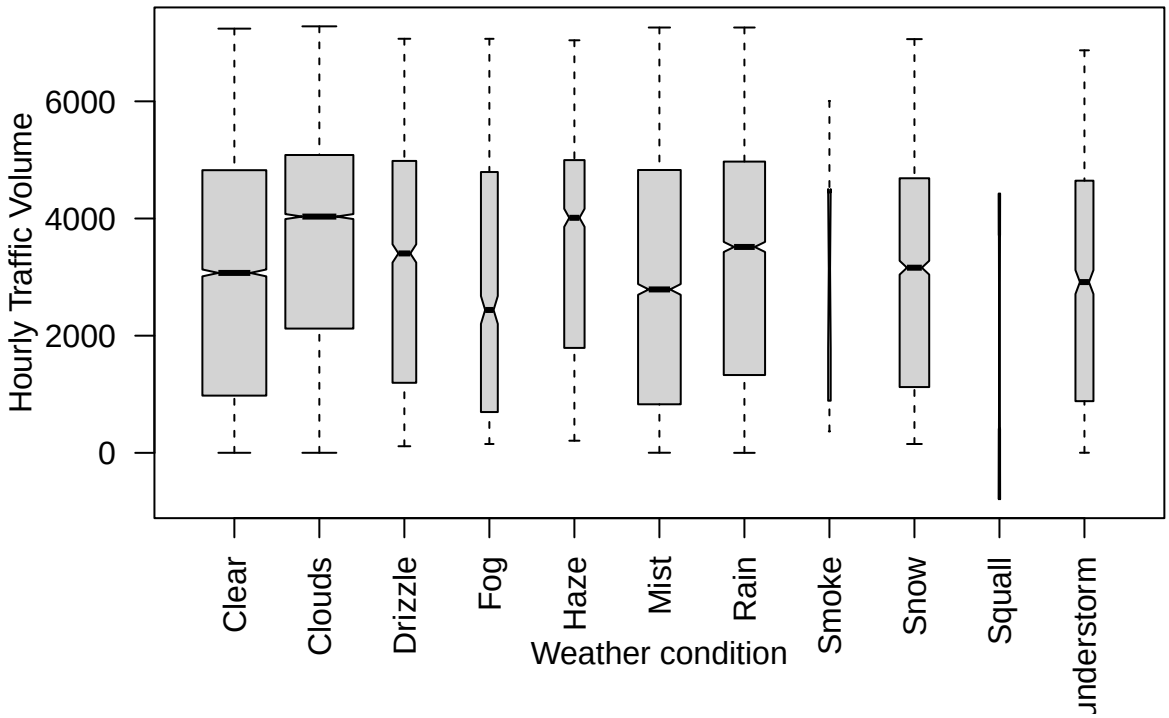
```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##       0    1196    3378    3261    4938    7280
```

Informative graphs of train data

```
# Removes rows with missing temperature data
train <- train[train$temp != 0, ]
# Creates a scatterplot of temperature vs. traffic volume
plot(train$temp, train$traffic_volume, pch='+', cex=0.75, col="blue", xlab="Temperature (kelvin)", ylab=
```




```
# Creates a boxplot of the traffic volume based on the weather condition
boxplot(train$traffic_volume~train$weather_main, varwidth=TRUE, notch=TRUE, xlab="Weather condition", yl
```



Simple linear regression model of train data

```
lm <- lm(traffic_volume~temp, data=train)
lm
```

```
##
## Call:
## lm(formula = traffic_volume ~ temp, data = train)
##
## Coefficients:
## (Intercept)         temp
##    -2441.72        20.28
```

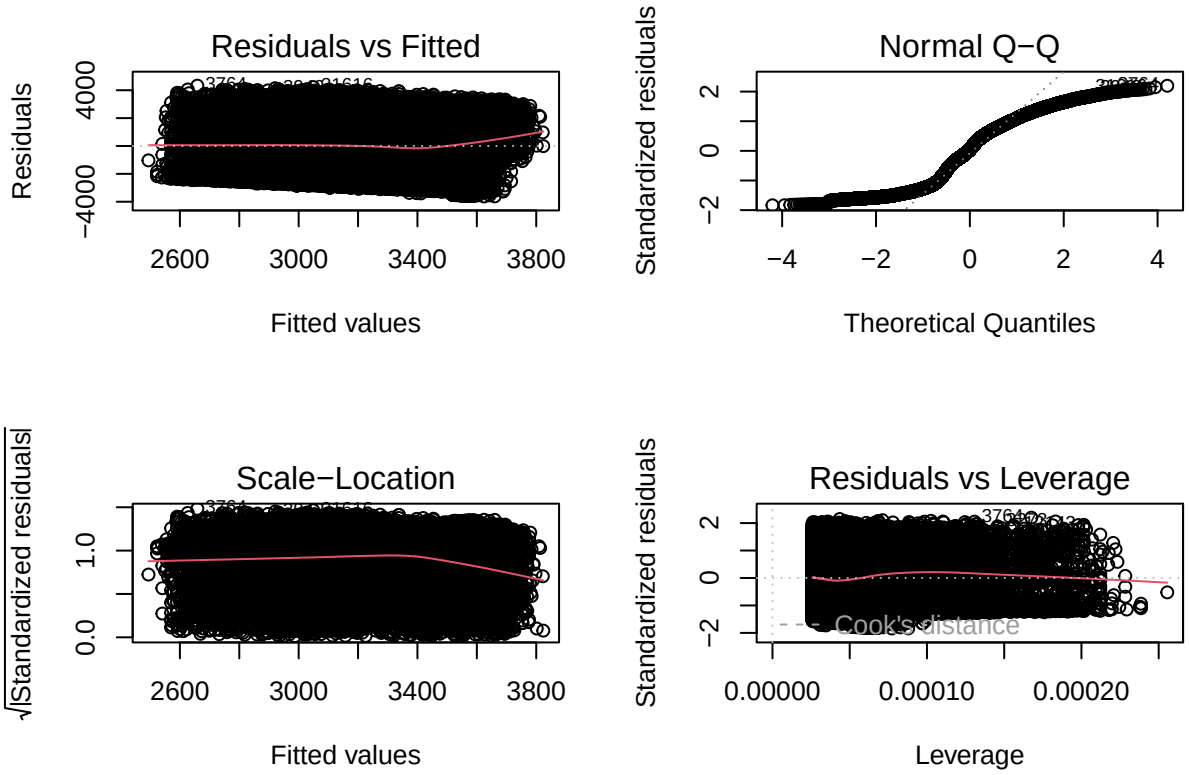
```
# Output the summary of the model
summary(lm)
```

```
##
## Call:
## lm(formula = traffic_volume ~ temp, data = train)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -3621.4 -1985.6    86.7  1666.3  4327.1
```

```
##
## Coefficients:
##               Estimate Std. Error t value Pr(>|t|)
## (Intercept) -2441.7181   222.1357  -10.99   <2e-16 ***
## temp           20.2798     0.7891   25.70   <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 1970 on 38554 degrees of freedom
## Multiple R-squared:  0.01684,    Adjusted R-squared:  0.01682
## F-statistic: 660.5 on 1 and 38554 DF,  p-value: < 2.2e-16
```

The residuals part of the summary displays information about the difference between the traffic volume predicted by the model and the actual traffic volume. We can see that the median difference between the predicted and actual traffic volumes is 86.7. The residual standard error shows how well a regression model fits the data. The smaller the residual standard error, the more accurate the model is. The residual standard error is fairly large for this model, indicating that the model underfits the data. R-squared represents the proportion of variance of y that can be explained by x in the model. In this case, that means that 1.68% of the variance in traffic volume can be explained by temperature in this model. This means that temperature is not the best predictor of traffic volume and that this model is not the best fit. The F-statistic shows how much the data varies from the mean. The F-statistic is very large in this case, meaning that the data differs largely from the mean.

```
# Plot the residuals of the model
par(mfrow=c(2,2))
plot(lm)
```



Multiple linear regression model of train data

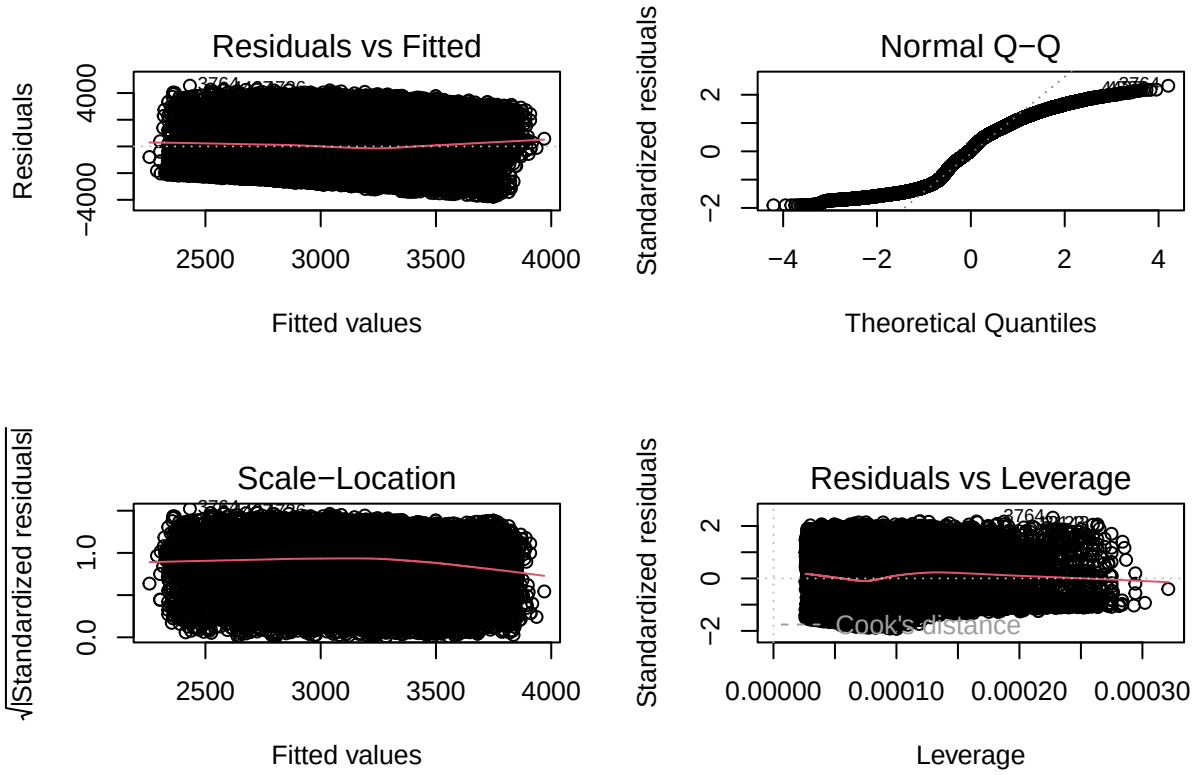
```
lm2 <- lm(traffic_volume~temp+clouds_all, data=train)
lm2
```

```
##
## Call:
## lm(formula = traffic_volume ~ temp + clouds_all, data = train)
##
## Coefficients:
## (Intercept)         temp   clouds_all
##   -3009.391       21.623        3.849
```

```
# Output the summary of the model
summary(lm2)
```

```
##
## Call:
## lm(formula = traffic_volume ~ temp + clouds_all, data = train)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -3746.9 -1945.1   105.1  1645.8  4553.2
##
## Coefficients:
##               Estimate Std. Error t value Pr(>|t|)
## (Intercept) -3009.3908   224.7518  -13.39   <2e-16 ***
## temp           21.6228     0.7920   27.30   <2e-16 ***
## clouds_all      3.8491     0.2582   14.90   <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 1965 on 38553 degrees of freedom
## Multiple R-squared:  0.02248,    Adjusted R-squared:  0.02243
## F-statistic: 443.2 on 2 and 38553 DF,  p-value: < 2.2e-16
```

```
# Plot the residuals of the model
par(mfrow=c(2,2))
plot(lm2)
```



Third linear regression model of train data

```
lm3 <- lm(traffic_volume~temp+clouds_all+rain_1h+snow_1h, data=train)
lm3
```

```
##
## Call:
## lm(formula = traffic_volume ~ temp + clouds_all + rain_1h + snow_1h,
##     data = train)
##
## Coefficients:
## (Intercept)         temp   clouds_all      rain_1h      snow_1h
##  -3006.6150      21.6129       3.8497       0.1429    -292.4057
```

```
# Output the summary of the model
summary(lm3)
```

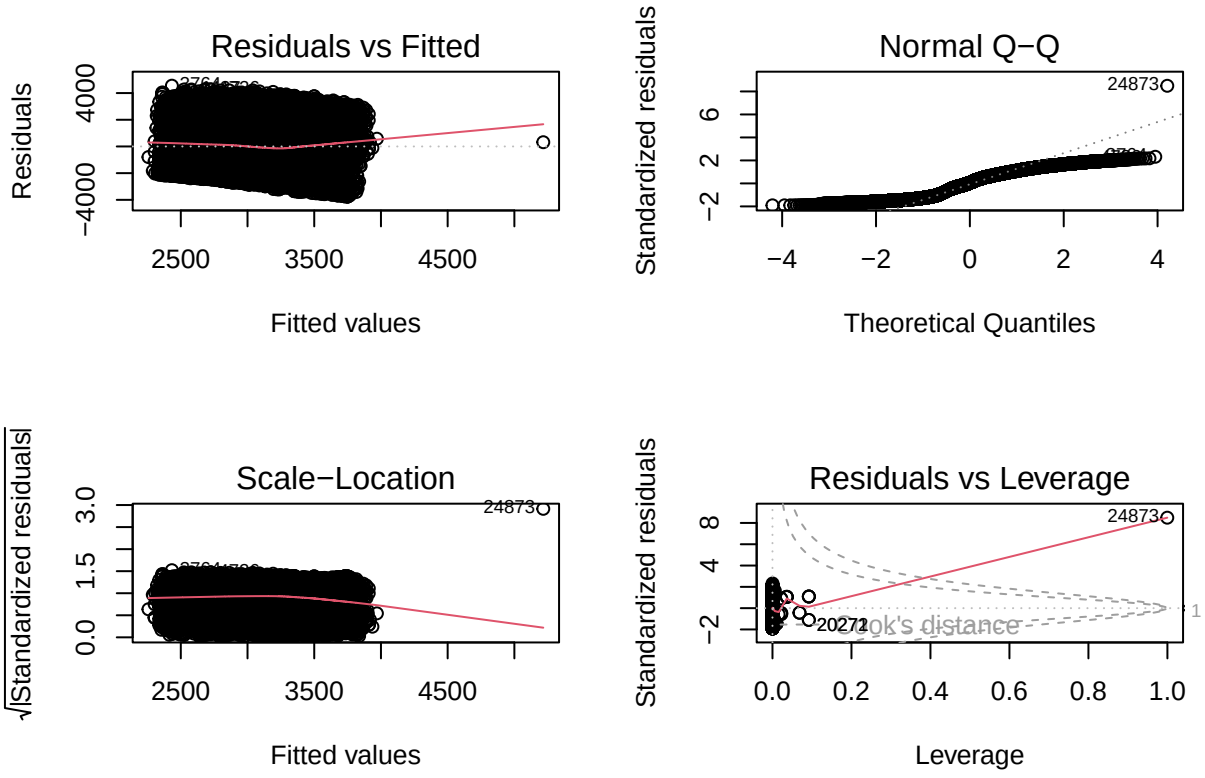
```
##
## Call:
## lm(formula = traffic_volume ~ temp + clouds_all + rain_1h + snow_1h,
##     data = train)
##
## Residuals:
```

```
##    Min     1Q Median     3Q    Max
##  -3747  -1945    105   1646   4553
##
## Coefficients:
##               Estimate Std. Error t value Pr(>|t|)
## (Intercept) -3006.6150   224.8092 -13.374   <2e-16 ***
## temp           21.6129     0.7922  27.283   <2e-16 ***
## clouds_all      3.8497     0.2583  14.901   <2e-16 ***
## rain_1h         0.1429     0.1998   0.715    0.475
## snow_1h      -292.4057  1170.1078  -0.250    0.803
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 1965 on 38551 degrees of freedom
## Multiple R-squared:  0.02249,    Adjusted R-squared:  0.02239
## F-statistic: 221.8 on 4 and 38551 DF,  p-value: < 2.2e-16
```

```r
# Plot the residuals of the model
par(mfrow=c(2,2))
plot(lm3)
```

```
## Warning in sqrt(crit * p * (1 - hh)/hh): NaNs produced

## Warning in sqrt(crit * p * (1 - hh)/hh): NaNs produced
```



Which model is the best?

The second model is the best based on the summaries of the three models. The residual standard error is the smallest in the second model compared to the other two. This means that the difference between the predicted and actual target is smaller in the second model. Therefore, the second regression model fits the data set the most accurately. The second model also has the larged R-squared value out of the three regression models. This means that this model predicts the target value the most accurately out of the three models.

Predict and evaluate on the test data

```
pred <- predict(lm, newdata=test)
cor <- cor(pred, test$traffic_volume)
mse <- mean((pred-test$traffic_volume)^2)
rmse <- sqrt(mse)
print(paste('correlation:', cor))
```

Model 1

```
## [1] "correlation: 0.141525726698018"
```

```
print(paste('mse:', mse))
```

```
## [1] "mse: 3863980.08432167"
```

```
print(paste('rmse:', rmse))
```

```
## [1] "rmse: 1965.70091425976"
```

```
pred2 <- predict(lm2, newdata=test)
cor2 <- cor(pred2, test$traffic_volume)
mse2 <- mean((pred2-test$traffic_volume)^2)
rmse2 <- sqrt(mse2)
print(paste('correlation:', cor2))
```

Model 2

```
## [1] "correlation: 0.176233246610858"
```

```
print(paste('mse:', mse2))
```

```
## [1] "mse: 3821511.54819976"
```

```
print(paste('rmse:', rmse2))
```

```
## [1] "rmse: 1954.86867799342"
```

```
pred3 <- predict(lm3, newdata=test)
cor3 <- cor(pred3, test$traffic_volume)
mse3 <- mean((pred3-test$traffic_volume)^2)
rmse3 <- sqrt(mse3)
print(paste('correlation:', cor3))
```

Model 3

```
## [1] "correlation: 0.176122519911885"
```

```
print(paste('mse:', mse3))
```

```
## [1] "mse: 3821656.66995747"
```

```
print(paste('rmse:', rmse3))
```

```
## [1] "rmse: 1954.90579567341"
```

Model 2 has the highest correlation. This can probably be explained by the lower residual standard error value of model 2, which indicates a better-fitting model.