R Tutorial Results

Thomas Higginbotham

# Prediction – Car Distance Traveled

In the first example, distance and speeds were provided. Using this data, I created a linear regression model that predicted the distances based on the provided speeds. The model seems to have performed better as the speeds increased:

|  |  |  |  |
| --- | --- | --- | --- |
| **Name** | **Provided Speed** | **Provided Distance** | **Predicted Distance** |
| Ford | 4 | 2 | -14.95415 |
| Jeep | 4 | 4 | -14.95415 |
| BMW | 9 | 16 | 10.41329 |
| GMC | 13 | 26 | 30.70724 |
| Chrysler | 13 | 28 | 30.70724 |
| Acura | 14 | 32 | 35.78073 |
| Chevrolet | 14 | 34 | 35.78073 |
| Buick | 14 | 34 | 35.78073 |
| Land Rover | 18 | 52 | 56.07468 |
| Lexus | 18 | 54 | 56.07468 |
| Nissan | 19 | 56 | 61.14817 |
| GMC | 20 | 60 | 66.22166 |
| Audi | 22 | 76 | 76.36864 |
| Buick | 24 | 84 | 86.51561 |
| Jeep | 24 | 85 | 86.51561 |

# Prediction – Petal Length

In the second example, petal length and width were provided. Using this data, I created a linear regression model that predicted the petal length based on the petal width.

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Sepal Length** | **Sepal Width** | **Petal Length** | **Petal Width** | **Species** | **Predicted Petal Width** |
| 5.1 | 3.5 | 1.4 | 0.2 | setosa | 0.2 |
| 4.9 | 3.0 | 1.4 | 0.2 | setosa | 0.2 |
| 4.7 | 3.2 | 1.3 | 0.2 | setosa | 0.2 |
| 5.0 | 3.6 | 1.4 | 0.2 | setosa | 0.2 |
| 5.4 | 3.7 | 1.5 | 0.2 | setosa | 0.3 |
| 5.1 | 3.5 | 1.4 | 0.3 | setosa | 0.2 |
| 5.7 | 3.8 | 1.7 | 0.3 | setosa | 0.4 |
| 5.2 | 3.5 | 1.5 | 0.2 | setosa | 0.3 |
| 5.2 | 3.4 | 1.4 | 0.2 | setosa | 0.2 |
| 5.2 | 4.1 | 1.5 | 0.1 | setosa | 0.3 |
| 5.0 | 3.2 | 1.2 | 0.2 | setosa | 0.2 |
| 5.1 | 3.8 | 1.9 | 0.4 | setosa | 0.4 |
| 4.6 | 3.2 | 1.4 | 0.2 | setosa | 0.2 |
| 5.3 | 3.7 | 1.5 | 0.2 | setosa | 0.3 |
| 6.5 | 2.8 | 4.6 | 1.5 | versicolor | 1.5 |
| 5.7 | 2.8 | 4.5 | 1.3 | versicolor | 1.5 |
| 6.3 | 3.3 | 4.7 | 1.6 | versicolor | 1.6 |
| 4.9 | 2.4 | 3.3 | 1.0 | versicolor | 1.0 |
| 6.6 | 2.9 | 4.6 | 1.3 | versicolor | 1.5 |
| 5.0 | 2.0 | 3.5 | 1.0 | versicolor | 1.1 |
| 5.9 | 3.0 | 4.2 | 1.5 | versicolor | 1.4 |
| 5.6 | 2.9 | 3.6 | 1.3 | versicolor | 1.1 |
| 6.7 | 3.1 | 4.4 | 1.4 | versicolor | 1.5 |
| 5.8 | 2.7 | 4.1 | 1.0 | versicolor | 1.3 |
| 5.6 | 2.5 | 3.9 | 1.1 | versicolor | 1.3 |
| 6.8 | 2.8 | 4.8 | 1.4 | versicolor | 1.6 |
| 5.8 | 2.7 | 3.9 | 1.2 | versicolor | 1.3 |
| 6.0 | 2.7 | 5.1 | 1.6 | versicolor | 1.7 |
| 5.0 | 2.3 | 3.3 | 1.0 | versicolor | 1.0 |
| 5.6 | 2.7 | 4.2 | 1.3 | versicolor | 1.4 |
| 6.2 | 2.9 | 4.3 | 1.3 | versicolor | 1.4 |
| 5.7 | 2.8 | 4.1 | 1.3 | versicolor | 1.3 |
| 6.3 | 3.3 | 6.0 | 2.5 | virginica | 2.1 |
| 6.3 | 2.9 | 5.6 | 1.8 | virginica | 2.0 |
| 6.5 | 3.0 | 5.8 | 2.2 | virginica | 2.0 |
| 6.5 | 3.2 | 5.1 | 2.0 | virginica | 1.7 |
| 6.8 | 3.0 | 5.5 | 2.1 | virginica | 1.9 |
| 6.4 | 3.2 | 5.3 | 2.3 | virginica | 1.8 |
| 6.7 | 3.3 | 5.7 | 2.1 | virginica | 2.0 |
| 7.4 | 2.8 | 6.1 | 1.9 | virginica | 2.2 |
| 6.4 | 2.8 | 5.6 | 2.2 | virginica | 2.0 |
| 6.1 | 2.6 | 5.6 | 1.4 | virginica | 2.0 |
| 6.9 | 3.1 | 5.4 | 2.1 | virginica | 1.9 |
| 6.7 | 3.1 | 5.6 | 2.4 | virginica | 2.0 |
| 6.7 | 3.3 | 5.7 | 2.5 | virginica | 2.0 |

# Errors and Warnings Encountered

* Error in install.packages : object 'readr' not found – use quotes around the package name
* Error in read.table(file = file, header = header, sep = sep, quote = quote, : object 'iris.csv' not found – Use quotes around the csv file name
* Error in summary(risDataset) : object 'risDataset' not found – Change ‘risDataset’ to ‘IrisDataset’
* Error in str(IrisDatasets) : object 'IrisDatasets' not found – Change ‘IrisDatasets’ to ‘IrisDataset’
* Error in hist.default(IrisDataset$Species) : 'x' must be numeric – Added new column as a factor of Species, then dropped the levels, and ran the histogram based on that new column.
* Error in FUN(X[[i]], ...) : only defined on a data frame with all numeric-alike variables – In qqnorm(), it required the value in parenthesis to be DatasetName$columnName.
* Error in eval(predvars, data, env) : object 'testingSet' not found – in the lm() function, the dataset is referenced at the end, with the dependent and independent variables as lm(independentVariable~ dependentVariable, trainingDataset)
* Error in predict(LinearModeltestSet) : object 'LinearModeltestSet' not found – to create a prediction, use predict(ModelName, testDataSet).
* Error: object 'predictions' not found – Check the name/spelling of the object

# Observations

Compared to setting up the Anaconda and Python environment, installing the latest version of R and RStudio seemed very easy. The tutorial was pretty direct and should be useful for users who already have some knowledge of Python.

With only one package loaded, linear regression models were created in R in fewer steps, and the validation and predictions seemed more direct than it would be in Python. The process of converting objects to factors, and having those levels represented by numbers, was also simple. That said, it’s important to understand the purpose of the model, get the training and test datasets sized correctly, understand the different structures that are being used, and the names of those structures and objects.

Employees who need to get started using R, and doing predictive analytics in R should plan to:

* Spend at least 20 hours completing tutorials such as this one, and getting familiar with the R commands and RStudio.
* Understand the basic aspects of linear regression models, and how to validate the performance of those models.
* Understand some of the basic error messages and warnings that might be encountered, and the strategies to identify the cause, the consequences, and corrective actions.