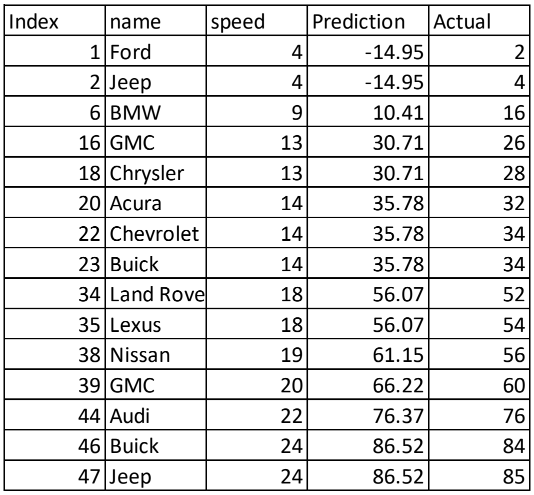
R and RStudio Informal Report - Course 3, Task 1

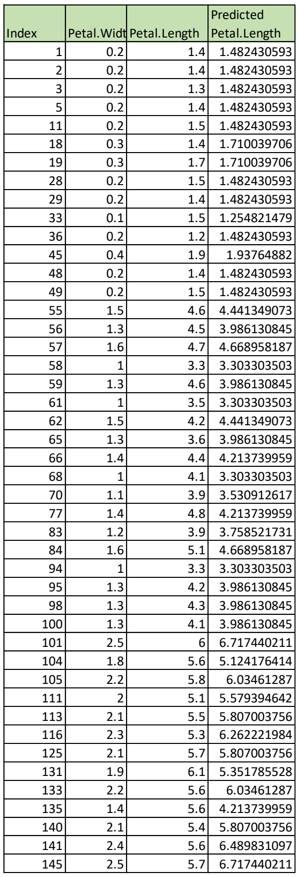
Car Tutorial - results



Index	name	speed	Prediction	Actual
1	Ford	4	-14.95	2
2	Jeep	4	-14.95	4
6	BMW	9	10.41	16
16	GMC	13	30.71	26
18	Chrysler	13	30.71	28
20	Acura	14	35.78	32
22	Chevrolet	14	35.78	34
23	Buick	14	35.78	34
34	Land Rove	18	56.07	52
35	Lexus	18	56.07	54
38	Nissan	19	61.15	56
39	GMC	20	66.22	60
44	Audi	22	76.37	76
46	Buick	24	86.52	84
47	Jeep	24	86.52	85

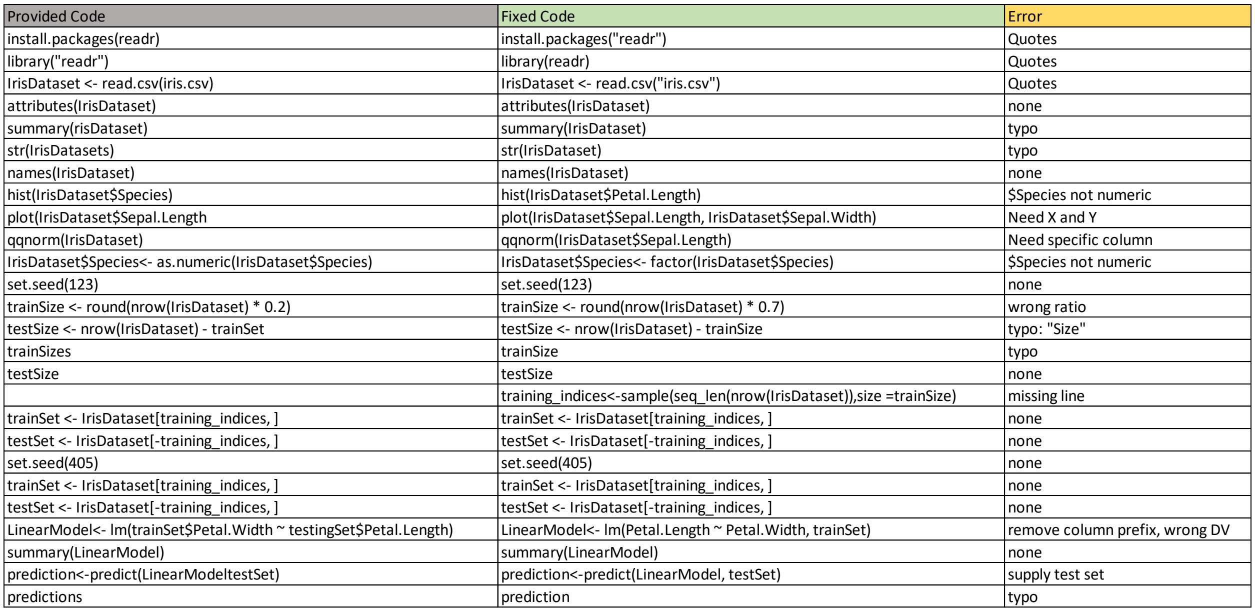
Figure 1: Cars test and predicted values

Iris Tutorial



Index	Petal.Widt	Petal.Length	Predicted Petal.Length
1	0.2	1.4	1.482430593
2	0.2	1.4	1.482430593
3	0.2	1.3	1.482430593
5	0.2	1.4	1.482430593
11	0.2	1.5	1.482430593
18	0.3	1.4	1.710039706
19	0.3	1.7	1.710039706
28	0.2	1.5	1.482430593
29	0.2	1.4	1.482430593
33	0.1	1.5	1.254821479
36	0.2	1.2	1.482430593
45	0.4	1.9	1.93764882
48	0.2	1.4	1.482430593
49	0.2	1.5	1.482430593
55	1.5	4.6	4.441349073
56	1.3	4.5	3.986130845
57	1.6	4.7	4.668958187
58	1	3.3	3.303303503
59	1.3	4.6	3.986130845
61	1	3.5	3.303303503
62	1.5	4.2	4.441349073
65	1.3	3.6	3.986130845
66	1.4	4.4	4.213739959
68	1	4.1	3.303303503
70	1.1	3.9	3.530912617
77	1.4	4.8	4.213739959
83	1.2	3.9	3.758521731
84	1.6	5.1	4.668958187
94	1	3.3	3.303303503
95	1.3	4.2	3.986130845
98	1.3	4.3	3.986130845
100	1.3	4.1	3.986130845
101	2.5	6	6.717440211
104	1.8	5.6	5.124176414
105	2.2	5.8	6.03461287
111	2	5.1	5.579394642
113	2.1	5.5	5.807003756
116	2.3	5.3	6.262221984
125	2.1	5.7	5.807003756
131	1.9	6.1	5.351785528
133	2.2	5.6	6.03461287
135	1.4	5.6	4.213739959
140	2.1	5.4	5.807003756
141	2.4	5.6	6.489831097
145	2.5	5.7	6.717440211

Figure 2: Iris test and predicted values



Provided Code	Fixed Code	Error
install.packages(readr)	install.packages("readr")	Quotes
library("readr")	library(readr)	Quotes
IrisDataset <- read.csv(iris.csv)	IrisDataset <- read.csv("iris.csv")	Quotes
attributes(IrisDataset)	attributes(IrisDataset)	none
summary(risDataset)	summary(IrisDataset)	typo
str(IrisDatasets)	str(IrisDataset)	typo
names(IrisDataset)	names(IrisDataset)	none
hist(IrisDataset$Species)	hist(IrisDataset$Petal.Length)	$Species not numeric
plot(IrisDataset$Sepal.Length	plot(IrisDataset$Sepal.Length, IrisDataset$Sepal.Width)	Need X and Y
qqnorm(IrisDataset)	qqnorm(IrisDataset$Sepal.Length)	Need specific column
IrisDataset$Species<- as.numeric(IrisDataset$Species)	IrisDataset$Species<- factor(IrisDataset$Species)	$Species not numeric
set.seed(123)	set.seed(123)	none
trainSize <- round(nrow(IrisDataset) * 0.2)	trainSize <- round(nrow(IrisDataset) * 0.7)	wrong ratio
testSize <- nrow(IrisDataset) - trainSet	testSize <- nrow(IrisDataset) - trainSize	typo: "Size"
trainSizes	trainSize	typo
testSize	testSize	none
	training_indices<-sample(seq_len(nrow(IrisDataset)),size =trainSize)	missing line
trainSet <- IrisDataset[training_indices,]	trainSet <- IrisDataset[training_indices,]	none
testSet <- IrisDataset[-training_indices,]	testSet <- IrisDataset[-training_indices,]	none
set.seed(405)	set.seed(405)	none
trainSet <- IrisDataset[training_indices,]	trainSet <- IrisDataset[training_indices,]	none
testSet <- IrisDataset[-training_indices,]	testSet <- IrisDataset[-training_indices,]	none
LinearModel<- lm(trainSet$Petal.Width ~ testingSet$Petal.Length)	LinearModel<- lm(Petal.Length ~ Petal.Width, trainSet)	remove column prefix, wrong DV
summary(LinearModel)	summary(LinearModel)	none
prediction<-predict(LinearModeltestSet)	prediction<-predict(LinearModel, testSet)	supply test set
predictions	prediction	typo

Figure 3: Script errors and fixes

Script errors consisted of syntax errors, variable name typos, incorrectly specifying columns for plots, defining *training_indices*, incorrect dependent variable, and other function calling errors.

Commentary

Installation of R and RStudio was straightforward. I did get some initial warnings when loading in the "readr" package, and to satisfy them, I also installed RTools, which was straightforward. The tutorial was straightforward, and the ability to debug script code was a very useful practice. I also played with the GUI import functions, and those seem to have a lot of good options for dealing with imperfect data (to an extent). The ability to sort the data before importing gives a good opportunity to spot bad data, as well as the ability to customize the column data types. It seems to come with the drawback of not being automatable for re-running data like a script.

The main lessons learned – it is fairly straightforward to import simple data sets, run basic descriptor queries on them, and train new models in RStudio. The errors returned for bad commands also tend to be more focused and descriptive than the errors normally seen in Jupyter notebook code.

One of the main benefits of RStudio is that it is truly open source, whereas Rapidminer limits functionality above a certain point before wanting a license. I don't know Rapidminer very well, but Rstudio feels pretty intuitive after completing this tutorial, and would be easy to recommend to others.