STA 545 ASSIGNMENT #1

Suruchi Jaikumar Ahuja

1) Statistical learning combines the field of statistics and functional analysis. It has two major divisions- supervised learning (involves learning and understanding of training set of data) and unsupervised learning.

a) Real- life applications in which Classification is useful

--*Handwriting recognition*- Here data is represented as vector sequences. The input data is of various types, it can be online or handwritten script, cursive script, uncontained script, isolated characters and so on. The input data has to be preprocessed first and the unwanted data is cleaned which might create a negative impact on the output. Then feature extraction is performed where high dimensional data is extracted. Then comes classification where the mapping of extracted features is done using models such as bagged classification trees, Hidden Markov model, n-grams(also used for text classification) for identifying the words and characters of the features. The goal of this handwriting recognition algorithm will be the inference of the data that is inputted for recognition.

--*Biometric Recognition* using statistical approach and the neural network methodology has been gaining a lot of attention recently. The designing of this recognition system involves use of classes, the sensing environment, feature extraction and selection, cluster analysis, selection of training and test samples and the performance evaluation. The biometric recognition is used for personal identification and the goal is for the prediction of identity of the individual. The input pattern can be the face, iris of the eye, fingerprint. The predefined pattern classes will include authorized users for access control.

Example**-** the iPhone 5s and later models use this pattern recognition technique where your iPhone can be unlocked with your fingerprint which is pre-fed in the system.

--*Medical Imaging* using statistical methods is being used to detect minute differences between a normal functional brain and a brain affected with dysfunction. It is a combination of Machine learning, pattern recognition and computer vision. A group analysis is performed which gives general effects and then a classification gives subject level measures and gives biomarkers for diagnosis. It gives smaller size subset and the goal is to interpret the inference as to which parts of anatomy are affected by that disease.

b) Applications of Regression Analysis in Real World

--Regression Analysis can be used for *Total quality management* where the effect of one variable can be seen on the other variable, in this case the effect of price on demand and vice versa. Using regression analysis prediction of the events which are still to occur is possible like demand analysis for a certain product. Once the model or the analysis is performed it has to be optimized with the use of statistical tools.

--The linear regression model analysis can be used in *Education* where the main goal is to predict the dependent variable, the students grade on an a test or a quiz using the independent variables such as the teachers, teaching methodology, family background and the schooling environment.

--The regression model can also be used in *Economics* using the independent variables such as family's income, number of children in the house, pets, number of people earning in the family, and the dependent variable that is the family's consumption expenditure can be estimated.

c) Use of cluster analysis

-- In *Bioinformatics and Computational Biology*, cluster analysis can be used for sequence analysis and genetic clustering. Hierarchical clustering builds a multilevel hierarchy of clusters by creating a cluster tree.

--Cluster Analysis is widely used in *Market research and grouping of shopping items in a shopping website*. The classification of customers, potential customers, product positioning and selecting the test market, identifying customer bases is done using cluster analysis.

--*Information Retrieval and the World Wide Web* uses clustering analysis, where the search results are grouped into small clusters which captures a particular aspect of a query. Each cluster will have a hierarchical structure of subcategories that further help a user in getting the wanted result.

2)

First the ElemStatLearn package is installed in the R console and the list of objects that is included in the library can be checked using the ls command. The zip test data and the zip train data are extracted from the package and are converted from the matrix form into a data frame test set using the conversion function (as.data.frame).

In the question, the first part says concentrate on digits 2 and 3, so here I made two new data frame sets called zip.test.23 and zip.train.23 where the previously stored testing data and training data is modified by attaching the 2’s and 3’s to the test set. Using the compact operator, $ sign we attach the V1 column containing the 2 and 3 in the test and training data frame. In the next step, I performed the k nearest neighbor method. The k-nn algorithm is implemented in the package class and that library has to be called. This method is for the test set from the training set. The k-nn is an easy to implement algorithm and the values of k can be changed to see how the performance changes. Here the k value is taken as 1, 3, 5, 7, 15. The k.error will calculate the residual square error. A for loop is used where the i variable starts from 1 and continues to iterate for the values of k=1, 3, 5, 7, 15. Then a KNN variable is taken which stores the calculated results using the knn function. It is calculated on the the zip.train.23 and the zip.test.23 (modified, but the last column is removed), and zip.train.23 with the v1 column attached to it.

A y_true and a y_test value are taken which will convert the knn value in numeric format. The knn error is calculated using the value (1/length(y_test))*sum(abs(y_test-y_true)).

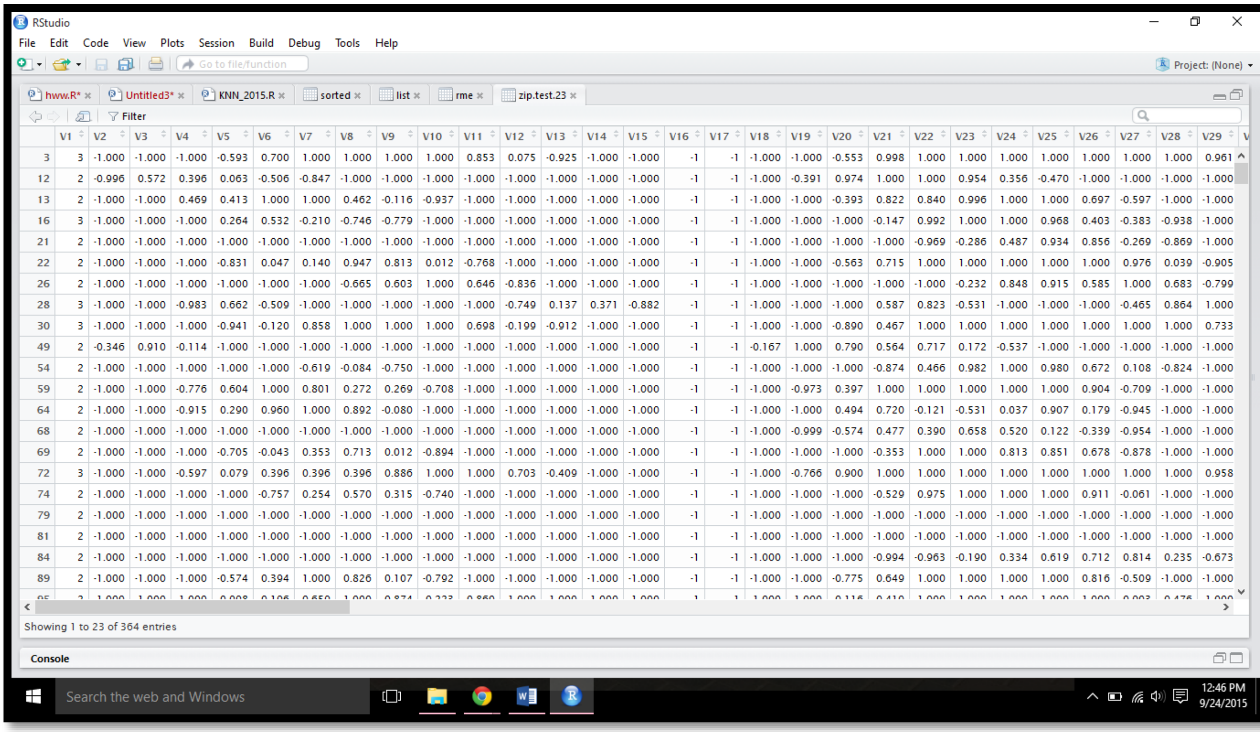
The linear regression, a linear model is built for the training set data. The intercept column is added to the testing data set and the residual sum error is calculated.

OUTPUT:

The package consists of

```
[1] "bone"              "countries"        "galaxy"           "marketing"        "
mixture.example"
 [6] "nci"              "orange10.test"    "orange10.train"   "orange4.test"     "
orange4.train"
[11] "ozone"            "phoneme"          "prostate"         "SAheart"          "
simple.ridge"
[16] "spam"             "vowel.test"       "vowel.train"      "waveform"         "
waveform.test"
[21] "waveform.train"   "zip.test"         "zip.train"        "zip2image"
```

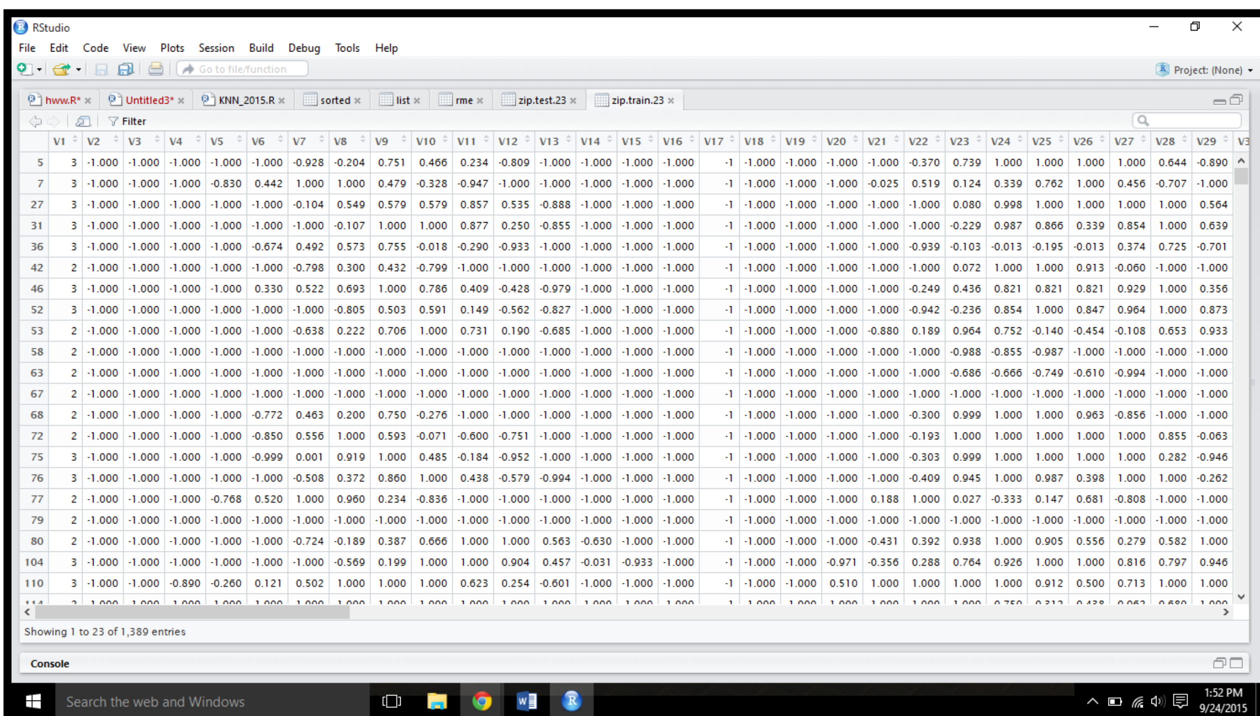
Figure 1-Modified Test set with 2's and 3's



	V1	V2	V3	V4	V5	V6	V7	V8	V9	V10	V11	V12	V13	V14	V15	V16	V17	V18	V19	V20	V21	V22	V23	V24	V25	V26	V27	V28	V29
3	3	-1.000	-1.000	-1.000	-0.593	0.700	1.000	1.000	1.000	1.000	0.853	0.075	-0.925	-1.000	-1.000	-1	-1	-1.000	-1.000	-0.553	0.998	1.000	1.000	1.000	1.000	1.000	1.000	1.000	0.961
12	2	-0.996	0.572	0.396	0.063	-0.506	-0.847	-1.000	-1.000	-1.000	-1.000	-1.000	-1.000	-1.000	-1.000	-1	-1	-1.000	-0.391	0.974	1.000	1.000	0.954	0.356	-0.470	-1.000	-1.000	-1.000	-1.000
13	2	-1.000	-1.000	0.469	0.413	1.000	1.000	0.462	-0.116	-0.937	-1.000	-1.000	-1.000	-1.000	-1.000	-1	-1	-1.000	-1.000	-0.393	0.822	0.840	0.996	1.000	1.000	0.697	-0.597	-1.000	-1.000
16	3	-1.000	-1.000	-1.000	0.264	0.532	-0.210	-0.746	-0.779	-1.000	-1.000	-1.000	-1.000	-1.000	-1.000	-1	-1	-1.000	-1.000	-1.000	-0.147	0.992	1.000	1.000	0.968	0.403	-0.383	-0.938	-1.000
21	2	-1.000	-1.000	-1.000	-1.000	-1.000	-1.000	-1.000	-1.000	-1.000	-1.000	-1.000	-1.000	-1.000	-1.000	-1	-1	-1.000	-1.000	-1.000	-1.000	-0.969	-0.286	0.487	0.934	0.856	-0.269	-0.869	-1.000
22	2	-1.000	-1.000	-1.000	-0.831	0.047	0.140	0.947	0.813	0.012	-0.768	-1.000	-1.000	-1.000	-1.000	-1	-1	-1.000	-1.000	-0.563	0.715	1.000	1.000	1.000	1.000	1.000	0.976	0.039	-0.905
26	2	-1.000	-1.000	-1.000	-1.000	-1.000	-1.000	-0.665	0.603	1.000	0.646	-0.836	-1.000	-1.000	-1.000	-1	-1	-1.000	-1.000	-1.000	-1.000	-1.000	-0.232	0.848	0.915	0.585	1.000	0.683	-0.799
28	3	-1.000	-1.000	-0.983	0.662	-0.509	-1.000	-1.000	-1.000	-1.000	-1.000	-0.749	0.137	0.371	-0.882	-1	-1	-1.000	-1.000	-1.000	0.587	0.823	-0.531	-1.000	-1.000	-1.000	-0.465	0.864	1.000
30	3	-1.000	-1.000	-1.000	-0.941	-0.120	0.858	1.000	1.000	1.000	0.698	-0.199	-0.912	-1.000	-1.000	-1	-1	-1.000	-1.000	-0.890	0.467	1.000	1.000	1.000	1.000	1.000	1.000	1.000	0.733
49	2	-0.346	0.910	-0.114	-1.000	-1.000	-1.000	-1.000	-1.000	-1.000	-1.000	-1.000	-1.000	-1.000	-1.000	-1	-1	-0.167	1.000	0.790	0.564	0.717	0.172	-0.537	-1.000	-1.000	-1.000	-1.000	-1.000
54	2	-1.000	-1.000	-1.000	-1.000	-1.000	-0.619	-0.084	-0.750	-1.000	-1.000	-1.000	-1.000	-1.000	-1.000	-1	-1	-1.000	-1.000	-1.000	-0.874	0.466	0.982	1.000	0.980	0.672	0.108	-0.824	-1.000
59	2	-1.000	-1.000	-0.776	0.604	1.000	0.801	0.272	0.269	-0.708	-1.000	-1.000	-1.000	-1.000	-1.000	-1	-1	-1.000	-0.973	0.397	1.000	1.000	1.000	1.000	1.000	0.904	-0.709	-1.000	-1.000
64	2	-1.000	-1.000	-0.915	0.290	0.960	1.000	0.892	-0.080	-1.000	-1.000	-1.000	-1.000	-1.000	-1.000	-1	-1	-1.000	-1.000	0.494	0.720	-0.121	-0.531	0.037	0.907	0.179	-0.945	-1.000	-1.000
68	2	-1.000	-1.000	-1.000	-1.000	-1.000	-1.000	-1.000	-1.000	-1.000	-1.000	-1.000	-1.000	-1.000	-1.000	-1	-1	-1.000	-0.999	-0.574	0.477	0.390	0.658	0.520	0.122	-0.339	-0.954	-1.000	-1.000
69	2	-1.000	-1.000	-1.000	-0.705	-0.043	0.353	0.713	0.012	-0.894	-1.000	-1.000	-1.000	-1.000	-1.000	-1	-1	-1.000	-1.000	-1.000	-0.353	1.000	1.000	0.813	0.851	0.678	-0.878	-1.000	-1.000
72	3	-1.000	-1.000	-0.597	0.079	0.396	0.396	0.396	0.886	1.000	1.000	0.703	-0.409	-1.000	-1.000	-1	-1	-1.000	-0.766	0.900	1.000	1.000	1.000	1.000	1.000	1.000	1.000	1.000	0.958
74	2	-1.000	-1.000	-1.000	-1.000	-0.757	0.254	0.570	0.315	-0.740	-1.000	-1.000	-1.000	-1.000	-1.000	-1	-1	-1.000	-1.000	-1.000	-0.529	0.975	1.000	1.000	1.000	0.911	-0.061	-1.000	-1.000
79	2	-1.000	-1.000	-1.000	-1.000	-1.000	-1.000	-1.000	-1.000	-1.000	-1.000	-1.000	-1.000	-1.000	-1.000	-1	-1	-1.000	-1.000	-1.000	-1.000	-1.000	-1.000	-1.000	-1.000	-1.000	-1.000	-1.000	-1.000
81	2	-1.000	-1.000	-1.000	-1.000	-1.000	-1.000	-1.000	-1.000	-1.000	-1.000	-1.000	-1.000	-1.000	-1.000	-1	-1	-1.000	-1.000	-1.000	-1.000	-1.000	-1.000	-1.000	-1.000	-1.000	-1.000	-1.000	-1.000
84	2	-1.000	-1.000	-1.000	-1.000	-1.000	-1.000	-1.000	-1.000	-1.000	-1.000	-1.000	-1.000	-1.000	-1.000	-1	-1	-1.000	-1.000	-1.000	-0.994	-0.963	-0.190	0.334	0.619	0.712	0.814	0.235	-0.673
89	2	-1.000	-1.000	-1.000	-0.574	0.394	1.000	0.826	0.107	-0.792	-1.000	-1.000	-1.000	-1.000	-1.000	-1	-1	-1.000	-1.000	-0.775	0.649	1.000	1.000	1.000	1.000	0.816	-0.509	-1.000	-1.000

Showing 1 to 23 of 364 entries

Figure 2- Modified Training Set with 2's and 3's.



	V1	V2	V3	V4	V5	V6	V7	V8	V9	V10	V11	V12	V13	V14	V15	V16	V17	V18	V19	V20	V21	V22	V23	V24	V25	V26	V27	V28	V29
5	3	-1.000	-1.000	-1.000	-1.000	-1.000	-0.928	-0.204	0.751	0.466	0.234	-0.809	-1.000	-1.000	-1.000	-1.000	-1	-1.000	-1.000	-1.000	-1.000	-0.370	0.739	1.000	1.000	1.000	1.000	0.644	-0.890
7	3	-1.000	-1.000	-1.000	-0.830	0.442	1.000	1.000	0.479	-0.328	-0.947	-1.000	-1.000	-1.000	-1.000	-1.000	-1	-1.000	-1.000	-1.000	-0.025	0.519	0.124	0.339	0.762	1.000	0.456	-0.707	-1.000
27	3	-1.000	-1.000	-1.000	-1.000	-1.000	-0.104	0.549	0.579	0.579	0.857	0.535	-0.888	-1.000	-1.000	-1.000	-1	-1.000	-1.000	-1.000	-1.000	-1.000	0.080	0.998	1.000	1.000	1.000	1.000	0.564
31	3	-1.000	-1.000	-1.000	-1.000	-1.000	-1.000	-0.107	1.000	1.000	0.877	0.250	-0.855	-1.000	-1.000	-1.000	-1	-1.000	-1.000	-1.000	-1.000	-1.000	-0.229	0.987	0.866	0.339	0.854	1.000	0.639
36	3	-1.000	-1.000	-1.000	-1.000	-0.674	0.492	0.573	0.755	-0.018	-0.290	-0.933	-1.000	-1.000	-1.000	-1.000	-1	-1.000	-1.000	-1.000	-1.000	-0.939	-0.103	-0.013	-0.195	-0.013	0.374	0.725	-0.701
42	2	-1.000	-1.000	-1.000	-1.000	-1.000	-0.798	0.300	0.432	-0.799	-1.000	-1.000	-1.000	-1.000	-1.000	-1.000	-1	-1.000	-1.000	-1.000	-1.000	-1.000	0.072	1.000	1.000	0.913	-0.060	-1.000	-1.000
46	3	-1.000	-1.000	-1.000	-1.000	0.330	0.522	0.693	1.000	0.786	0.409	-0.428	-0.979	-1.000	-1.000	-1.000	-1	-1.000	-1.000	-1.000	-1.000	-0.249	0.436	0.821	0.821	0.821	0.929	1.000	0.356
52	3	-1.000	-1.000	-1.000	-1.000	-1.000	-1.000	-0.805	0.503	0.591	0.149	-0.562	-0.827	-1.000	-1.000	-1.000	-1	-1.000	-1.000	-1.000	-1.000	-0.942	-0.236	0.854	1.000	0.847	0.964	1.000	0.873
53	2	-1.000	-1.000	-1.000	-1.000	-1.000	-0.638	0.222	0.706	1.000	0.731	0.190	-0.685	-1.000	-1.000	-1.000	-1	-1.000	-1.000	-1.000	-0.880	0.189	0.964	0.752	-0.140	-0.454	-0.108	0.653	0.933
58	2	-1.000	-1.000	-1.000	-1.000	-1.000	-1.000	-1.000	-1.000	-1.000	-1.000	-1.000	-1.000	-1.000	-1.000	-1.000	-1	-1.000	-1.000	-1.000	-1.000	-1.000	-0.988	-0.855	-0.987	-1.000	-1.000	-1.000	-1.000
63	2	-1.000	-1.000	-1.000	-1.000	-1.000	-1.000	-1.000	-1.000	-1.000	-1.000	-1.000	-1.000	-1.000	-1.000	-1.000	-1	-1.000	-1.000	-1.000	-1.000	-1.000	-0.686	-0.666	-0.749	-0.610	-0.994	-1.000	-1.000
67	2	-1.000	-1.000	-1.000	-1.000	-1.000	-1.000	-1.000	-1.000	-1.000	-1.000	-1.000	-1.000	-1.000	-1.000	-1.000	-1	-1.000	-1.000	-1.000	-1.000	-1.000	-1.000	-1.000	-1.000	-1.000	-1.000	-1.000	-1.000
68	2	-1.000	-1.000	-1.000	-1.000	-0.772	0.463	0.200	0.750	-0.276	-1.000	-1.000	-1.000	-1.000	-1.000	-1.000	-1	-1.000	-1.000	-1.000	-1.000	-0.300	0.999	1.000	1.000	0.963	-0.856	-1.000	-1.000
72	2	-1.000	-1.000	-1.000	-1.000	-0.850	0.556	1.000	0.593	-0.071	-0.600	-0.751	-1.000	-1.000	-1.000	-1.000	-1	-1.000	-1.000	-1.000	-1.000	-0.193	1.000	1.000	1.000	1.000	1.000	0.855	-0.063
75	3	-1.000	-1.000	-1.000	-1.000	-0.999	0.001	0.919	1.000	0.485	-0.184	-0.952	-1.000	-1.000	-1.000	-1.000	-1	-1.000	-1.000	-1.000	-1.000	-0.303	0.999	1.000	1.000	1.000	1.000	0.282	-0.946
76	3	-1.000	-1.000	-1.000	-1.000	-1.000	-0.508	0.372	0.860	1.000	0.438	-0.579	-0.994	-1.000	-1.000	-1.000	-1	-1.000	-1.000	-1.000	-1.000	-0.409	0.945	1.000	0.987	0.398	1.000	1.000	-0.262
77	2	-1.000	-1.000	-1.000	-0.768	0.520	1.000	0.960	0.234	-0.836	-1.000	-1.000	-1.000	-1.000	-1.000	-1.000	-1	-1.000	-1.000	-1.000	0.188	1.000	0.027	-0.333	0.147	0.681	-0.808	-1.000	-1.000
79	2	-1.000	-1.000	-1.000	-1.000	-1.000	-1.000	-1.000	-1.000	-1.000	-1.000	-1.000	-1.000	-1.000	-1.000	-1.000	-1	-1.000	-1.000	-1.000	-1.000	-1.000	-1.000	-1.000	-1.000	-1.000	-1.000	-1.000	-1.000
80	2	-1.000	-1.000	-1.000	-1.000	-1.000	-0.724	-0.189	0.387	0.666	1.000	1.000	0.563	-0.630	-1.000	-1.000	-1	-1.000	-1.000	-1.000	-0.431	0.392	0.938	1.000	0.905	0.556	0.279	0.582	1.000
104	3	-1.000	-1.000	-1.000	-1.000	-1.000	-1.000	-0.569	0.199	1.000	1.000	0.904	0.457	-0.031	-0.933	-1.000	-1	-1.000	-1.000	-0.971	-0.356	0.288	0.764	0.926	1.000	1.000	0.816	0.797	0.946
110	3	-1.000	-1.000	-0.890	-0.260	0.121	0.502	1.000	1.000	1.000	0.623	0.254	-0.601	-1.000	-1.000	-1.000	-1	-1.000	-1.000	0.510	1.000	1.000	1.000	1.000	0.912	0.500	0.713	1.000	1.000

Showing 1 to 23 of 1,389 entries

```
                          Error Rate
Linear Regression 0.04112103
K-nn with k= 1    0.02472527
K-nn with k= 3    0.03021978
K-nn with k= 5    0.03021978
K-nn with k= 7    0.03296703
K-nn with k= 15   0.03846154
```

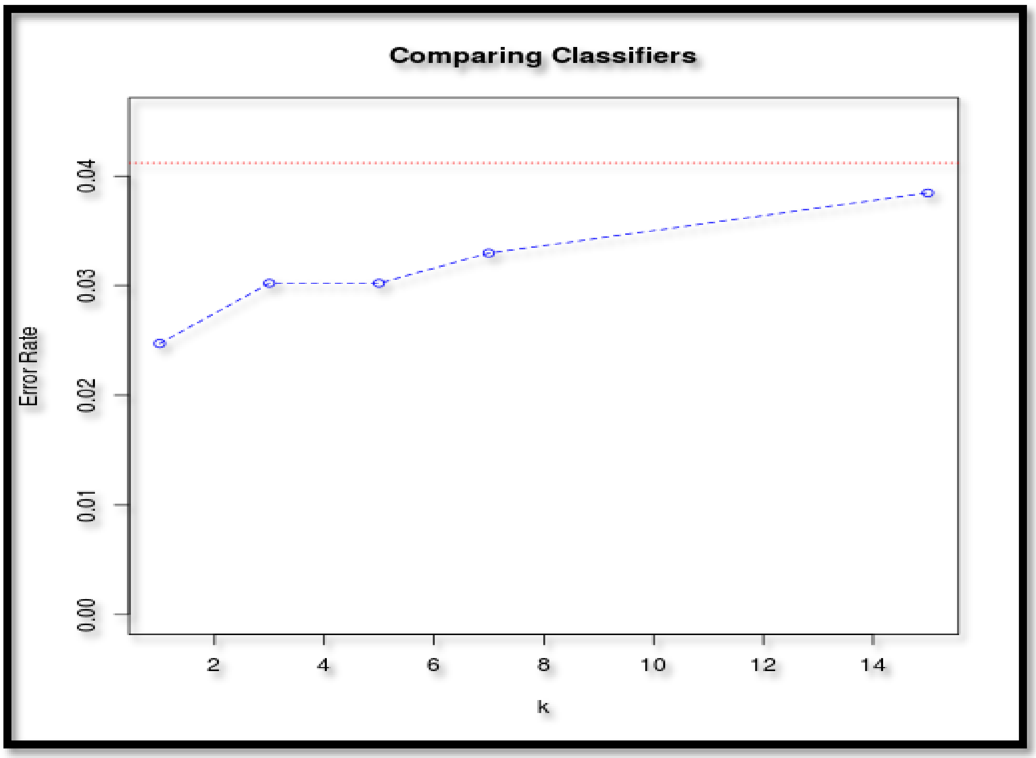


Figure 3 - The error rate of the linear regression classification is indicated by the red dotted line, and the blue dashed line gives the *k*-NN error rates for the different k values (1, 3, 5, 7, 15).

For the particular data set in this question, *k*-NN algorithm the small k values performs in a better way than linear regression. In the *k*-NN algorithms, the smaller k value is, the better the performance is. This occurs due to the "curse of dimensionality" problem i.e. as the number of dimensions increase the distances between the points increases and with 256 features. They are spread out so far that often their "nearest neighbors" aren't actually very near them.

3)

Here the library ggplot2 and MASS are installed and the data set Boston is called.

a) ***Pairwise scatterplots*** of the predictors are made using the pairs function.

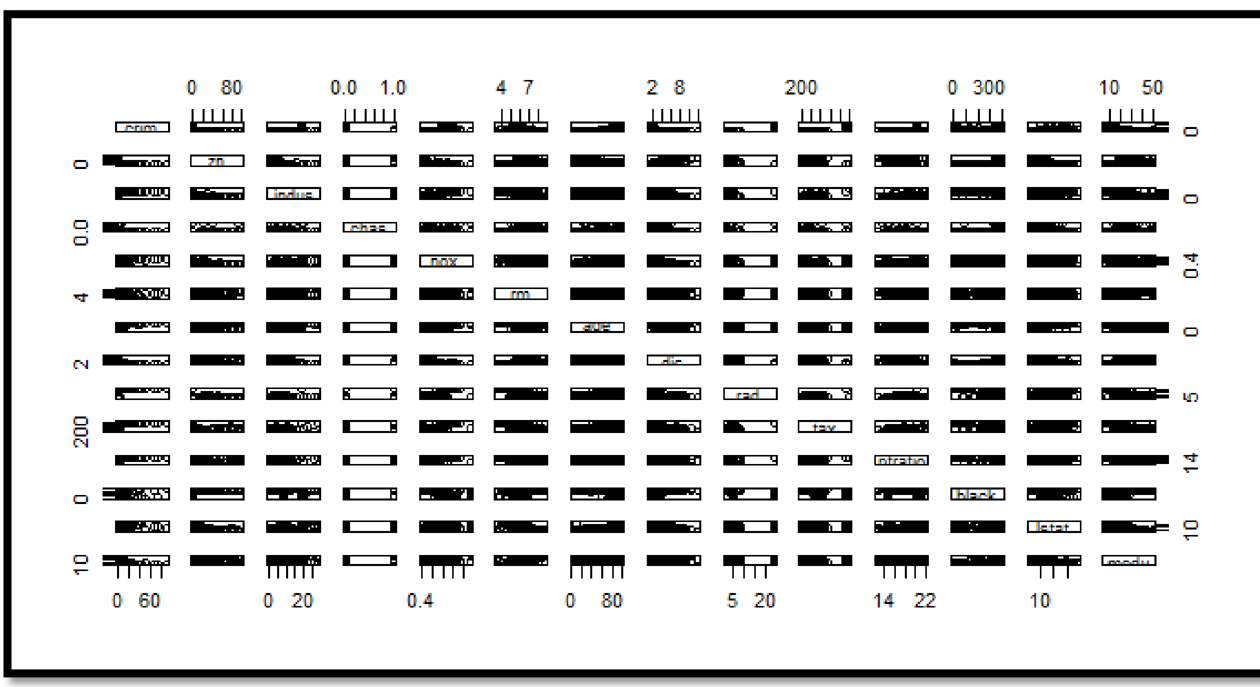


Figure 4- Pairwise Scatterplots of predictors of Boston dataset

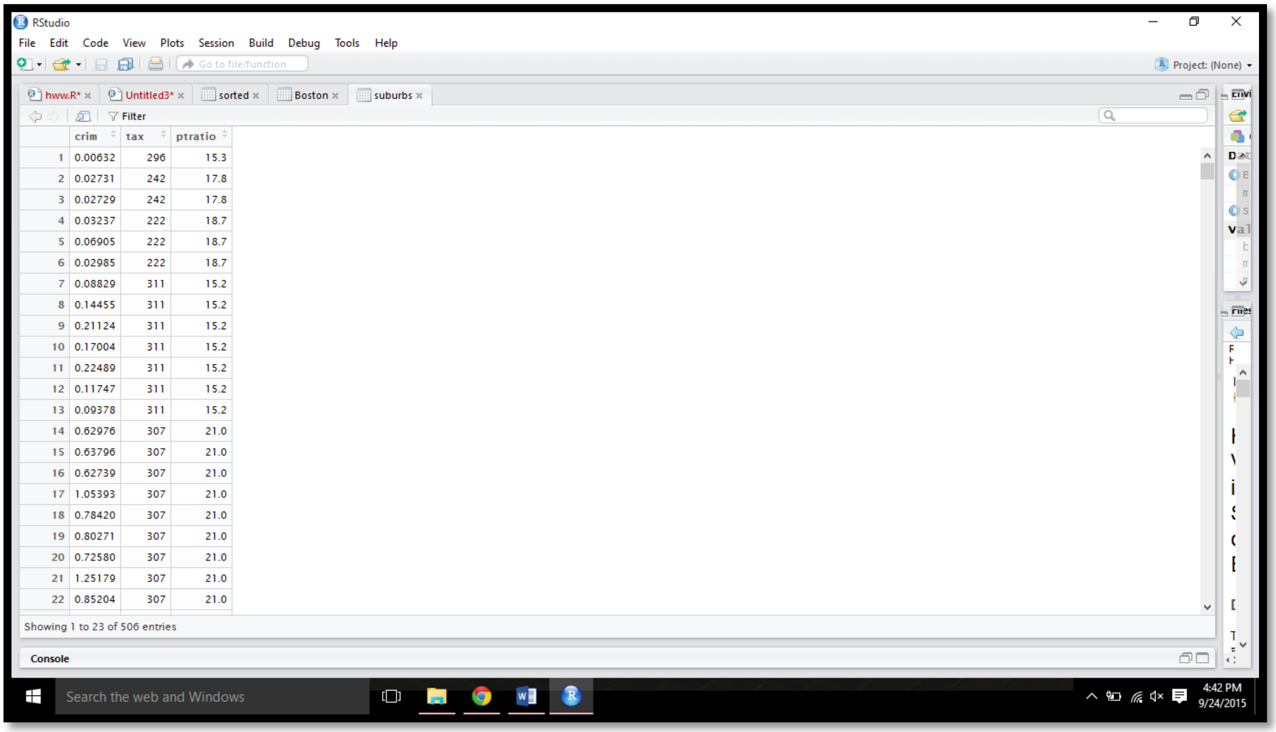
b)

The *covariance, variance and correlation* of each predictor in the Boston data set is checked with the per capita crime rate. The zn, chas, rm, dis, black, medv have negative covariance values. Negative value indicates, that the variables are negatively linearly related and the scatter plot almost falls along a straight line. It has a negative slope. A zero value, indicates a weak linear relationship between the variables. The indus, nox, age, rad, tax, ptratio, lstat have positive covariance values. The Positive value indicates that the variables are positively linearly related and the scatter plot falls almost along a straight line. It has a positive slope.

The highest covariance is shown by RAD (index of accessibility to r adial highways) with a value of 0.6255051 and since the value is close to 1, it is a positive variance and it is most linearly related to the per cap ita crime rate.

c) The Boston data set is attached in the console. The columns per capita crime rate, tax rate, and pupil-teacher ratios are extracted and stored in a new data set. Each column value is tested individually and compared wit h an assumed value. The assumed value is chosen as such it is closest to t he higher value in the range (like for example in the tax rate values are c ompared to 650, as the highest range is 711). The columns are then sorte d in descending order and stored together in one list using the cbind func tion. The range of each predictor is found using the range function.

OUTPUT:



	crim	tax	ptratio
1	0.00632	296	15.3
2	0.02731	242	17.8
3	0.02729	242	17.8
4	0.03237	222	18.7
5	0.06905	222	18.7
6	0.02985	222	18.7
7	0.08829	311	15.2
8	0.14455	311	15.2
9	0.21124	311	15.2
10	0.17004	311	15.2
11	0.22489	311	15.2
12	0.11747	311	15.2
13	0.09378	311	15.2
14	0.62976	307	21.0
15	0.63796	307	21.0
16	0.62739	307	21.0
17	1.05393	307	21.0
18	0.78420	307	21.0
19	0.80271	307	21.0
20	0.72580	307	21.0
21	1.25179	307	21.0
22	0.85204	307	21.0

Figure 5: This contains the list of the three columns crim, tax and ptratio in descending order.

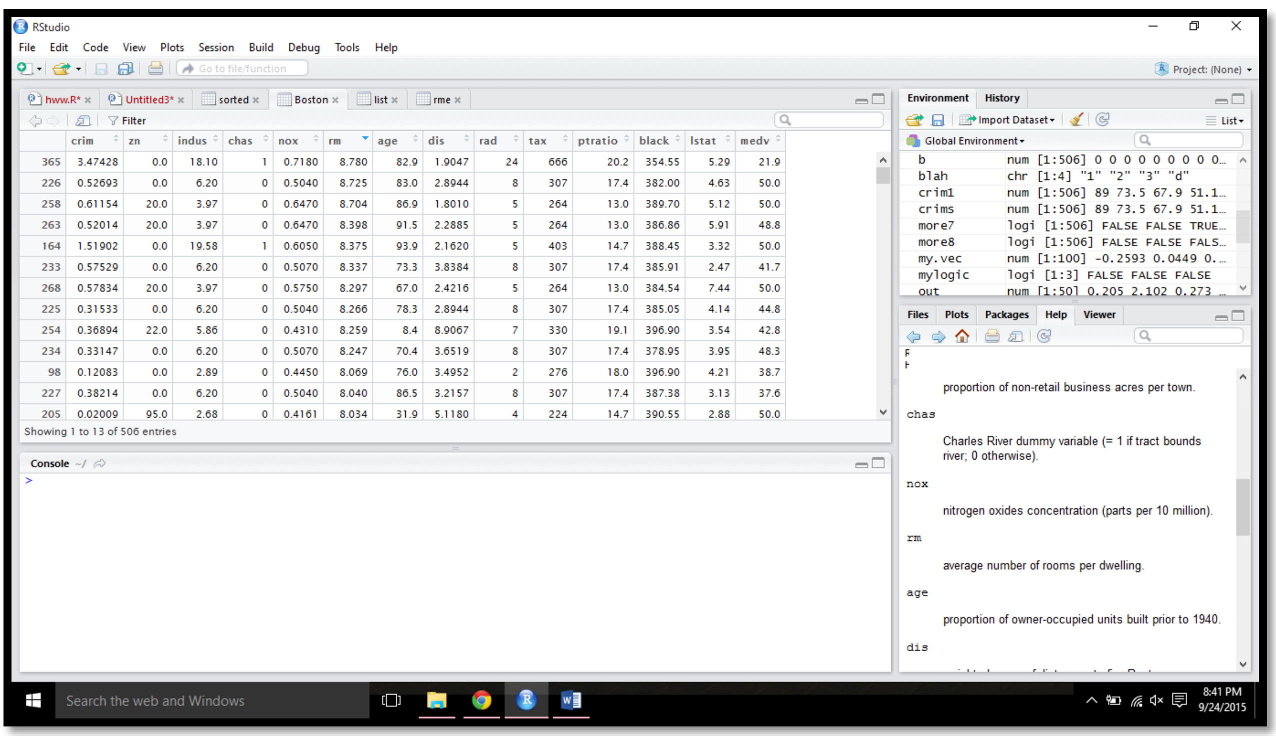
```
> range(crim)
[1]  0.00632 88.97620
> range(tax)
[1] 187 711
> range(ptratio)
[1] 12.6 22.0
```

The range of crim is found and the lowest value is 0.00632 and the highe st values is 88.97620, so all the values of crim will lie between these two values. Similarly the range of tax is from 187 to 711 and all the values w ill lie between these two values and ptratio is from 12.6 to 22.

d)

The number of suburbs that average more than seven room per d welling are 64. The number of suburbs that average more than ei ght room per dwelling are 13.

The suburbs containing more than 8 rooms per dwelling have the age predictor (proportion of owner-occupied units built prior to 1940.) on the higher side, most of the other predictors also are in a quite similar to each other with a few exceptions.



	crim	zn	indus	chas	nox	rm	age	dis	rad	tax	ptratio	black	lstat	medv
365	3.47428	0.0	18.10	1	0.7180	8.780	82.9	1.9047	24	666	20.2	354.55	5.29	21.9
226	0.52693	0.0	6.20	0	0.5040	8.725	83.0	2.8944	8	307	17.4	382.00	4.63	50.0
258	0.61154	20.0	3.97	0	0.6470	8.704	86.9	1.8010	5	264	13.0	389.70	5.12	50.0
263	0.52014	20.0	3.97	0	0.6470	8.398	91.5	2.2885	5	264	13.0	386.86	5.91	48.8
164	1.51902	0.0	19.58	1	0.6050	8.375	93.9	2.1620	5	403	14.7	388.45	3.32	50.0
233	0.57529	0.0	6.20	0	0.5070	8.337	73.3	3.8384	8	307	17.4	385.91	2.47	41.7
268	0.57834	20.0	3.97	0	0.5750	8.297	67.0	2.4216	5	264	13.0	384.54	7.44	50.0
225	0.31533	0.0	6.20	0	0.5040	8.266	78.3	2.8944	8	307	17.4	385.05	4.14	44.8
254	0.36894	22.0	5.86	0	0.4310	8.259	8.4	8.9067	7	330	19.1	396.90	3.54	42.8
234	0.33147	0.0	6.20	0	0.5070	8.247	70.4	3.6519	8	307	17.4	378.95	3.95	48.3
98	0.12083	0.0	2.89	0	0.4450	8.069	76.0	3.4952	2	276	18.0	396.90	4.21	38.7
227	0.38214	0.0	6.20	0	0.5040	8.040	86.5	3.2157	8	307	17.4	387.38	3.13	37.6
205	0.02009	95.0	2.68	0	0.4161	8.034	31.9	5.1180	4	224	14.7	390.55	2.88	50.0

---THE END---
