**STA 546 – Homework #1**

**20-February-2016 Suruchi Jaikumar Ahuja**

1. Question 1

The bodyfat dataset has measurements from an investigator who is interested in finding relationships between the different variables, especially related to the bodyfat measures.

The bodyfat dataset is loaded in the R global environment it has 252 observations of 15 variables. The variables are density, bodyfat, age, weight, height, neck, chest, abdomen, hips, thigh, knee, ankle, biceps, forearm and wrist.

Using the is.na() function, missing values can be checked for in the dataset. This dataset does not contain any missing values.

Histogram was plotted to check for the unusual distributions in the data. The data is distributed is not uniform and equal.

Boxplot is a convenient way of graphically depicting groups of numerical data through their [quartiles](https://en.wikipedia.org/wiki/Quartile). Box plots may also have lines extending vertically from the boxes indicating variability outside the upper and lower quartiles, hence the terms box-and-whisker plot and box-and-whisker diagram. [Outliers](https://en.wikipedia.org/wiki/Outlier) may be plotted as individual points.  An outlier is an observation point that is distant from other observations. An outlier may be due to variability in the measurement or it may indicate experimental error; the latter are sometimes excluded from the [data set](https://en.wikipedia.org/wiki/Data_set). Outliers can occur by chance in any distribution, but they often indicate either [measurement error](https://en.wikipedia.org/wiki/Measurement_error) or that the population has a [heavy-tailed distribution](https://en.wikipedia.org/wiki/Heavy-tailed_distribution). Boxplots were plotted to check for outliers in the dataset.

The data was then created into subsets to remove the outliers. After carefully removing the subsets the number of outliers have drastically reduced.

The data in this did not have scaling issues and scaling of the data wasn’t required.

Identify outliers using the boxplot.stats function

> boxplot.stats(data[["density"]],coef=1.5)

$stats

[1] 1.0101 1.0414 1.0549 1.0704 1.1089

$n

[1] 252

$conf

[1] 1.052014 1.057786

$out

[1] 0.995

> boxplot.stats(data[["bodyfat"]],coef=1.5)

$stats

[1] 0.00 12.45 19.20 25.30 40.10

$n

[1] 252

$conf

[1] 17.92103 20.47897

$out

[1] 47.5

> boxplot.stats(data[["age"]],coef=1.5)

$stats

[1] 22.0 35.5 43.0 54.0 81.0

$n

[1] 252

$conf

[1] 41.15868 44.84132

$out

integer(0)

> boxplot.stats(data[["weight"]],coef=1.5)

$stats

[1] 118.50 158.75 176.50 197.00 247.25

$n

[1] 252

$conf

[1] 172.693 180.307

$out

[1] 363.15 262.75

> boxplot.stats(data[["height"]],coef=1.5)

$stats

[1] 64.00 68.25 70.00 72.25 77.75

$n

[1] 252

$conf

[1] 69.60188 70.39812

$out

numeric(0)

> boxplot.stats(data[["neck"]],coef=1.5)

$stats

[1] 32.80 36.40 38.00 39.45 43.90

$n

[1] 252

$conf

[1] 37.69643 38.30357

$out

[1] 51.2 31.5 31.1

> boxplot.stats(data[["chest"]],coef=1.5)

$stats

[1] 79.30 94.30 99.65 105.45 121.60

$n

[1] 252

$conf

[1] 98.54023 100.75977

$out

[1] 136.2 128.3

> boxplot.stats(data[["abdomen"]],coef=1.5)

$stats

[1] 69.40 84.55 90.95 99.45 118.00

$n

[1] 252

$conf

[1] 89.46699 92.43301

$out

[1] 148.1 126.2 122.1

> boxplot.stats(data[["hip"]],coef=1.5)

$stats

[1] 85.00 95.50 99.30 103.55 115.50

$n

[1] 252

$conf

[1] 98.49878 100.10122

$out

[1] 116.1 147.7 125.6

> boxplot.stats(data[["thigh"]],coef=1.5)

$stats

[1] 47.2 56.0 59.0 62.4 71.2

$n

[1] 252

$conf

[1] 58.363 59.637

$out

[1] 87.3 72.5 72.9 74.4

> boxplot.stats(data[["knee"]],coef=1.5)

$stats

[1] 33.00 36.95 38.50 39.95 44.20

$n

[1] 252

$conf

[1] 38.20141 38.79859

$out

[1] 49.1 45.0 46.0

> boxplot.stats(data[["ankle"]],coef=1.5)

$stats

[1] 19.1 22.0 22.8 24.0 27.0

$n

[1] 252

$conf

[1] 22.60094 22.99906

$out

[1] 33.9 29.6 33.7

> boxplot.stats(data[["biceps"]],coef=1.5)

$stats

[1] 24.80 30.20 32.05 34.35 39.10

$n

[1] 252

$conf

[1] 31.63695 32.46305

$out

[1] 45

> boxplot.stats(data[["forearm"]],coef=1.5)

$stats

[1] 24.6 27.3 28.7 30.0 33.8

$n

[1] 252

$conf

[1] 28.43127 28.96873

$out

[1] 23.1 34.9 21.0 23.1 22.0

> boxplot.stats(data[["wrist"]],coef=1.5)

$stats

[1] 16.1 17.6 18.3 18.8 20.4

$n

[1] 252

$conf

[1] 18.18056 18.41944

$out

[1] 21.4 21.4 15.8 20.9

Checks the correlation of each variable in the dataset

> cor(data)

density bodyfat age weight height neck chest abdomen

density 1.00000000 -0.99909226 -0.28952932 -0.61315218 0.01928952 -0.4909661 -0.7026080 -0.8123356

bodyfat -0.99909226 1.00000000 0.29145844 0.61241400 -0.02528683 0.4905919 0.7026203 0.8134323

age -0.28952932 0.29145844 1.00000000 -0.01274609 -0.24521233 0.1135052 0.1764497 0.2304094

weight -0.61315218 0.61241400 -0.01274609 1.00000000 0.48688800 0.8307162 0.8941905 0.8879949

height 0.01928952 -0.02528683 -0.24521233 0.48688800 1.00000000 0.3211409 0.2268286 0.1897662

neck -0.49096607 0.49059185 0.11350519 0.83071622 0.32114085 1.0000000 0.7848350 0.7540774

chest -0.70260796 0.70262034 0.17644968 0.89419052 0.22682861 0.7848350 1.0000000 0.9158277

abdomen -0.81233562 0.81343228 0.23040942 0.88799494 0.18976623 0.7540774 0.9158277 1.0000000

hip -0.62495959 0.62520092 -0.05033212 0.94088412 0.37210602 0.7349579 0.8294199 0.8740662

thigh -0.56377862 0.55960753 -0.20009576 0.86869354 0.33855758 0.6956973 0.7298586 0.7666239

knee -0.51173330 0.50866524 0.01751569 0.85316739 0.50050052 0.6724050 0.7194964 0.7371789

ankle -0.26796851 0.26596977 -0.10505810 0.61368542 0.39313147 0.4778924 0.4829879 0.4532227

biceps -0.49479036 0.49327113 -0.04116212 0.80041593 0.31850749 0.7311459 0.7279075 0.6849827

forearm -0.36499308 0.36138690 -0.08505555 0.63030143 0.32202734 0.6236603 0.5801727 0.5033161

wrist -0.34906981 0.34657486 0.21353062 0.72977489 0.39777960 0.7448264 0.6601623 0.6198324

hip thigh knee ankle biceps forearm wrist

density -0.62495959 -0.5637786 -0.51173330 -0.2679685 -0.49479036 -0.36499308 -0.3490698

bodyfat 0.62520092 0.5596075 0.50866524 0.2659698 0.49327113 0.36138690 0.3465749

age -0.05033212 -0.2000958 0.01751569 -0.1050581 -0.04116212 -0.08505555 0.2135306

weight 0.94088412 0.8686935 0.85316739 0.6136854 0.80041593 0.63030143 0.7297749

height 0.37210602 0.3385576 0.50050052 0.3931315 0.31850749 0.32202734 0.3977796

neck 0.73495788 0.6956973 0.67240498 0.4778924 0.73114592 0.62366027 0.7448264

chest 0.82941992 0.7298586 0.71949640 0.4829879 0.72790748 0.58017273 0.6601623

abdomen 0.87406618 0.7666239 0.73717888 0.4532227 0.68498272 0.50331609 0.6198324

hip 1.00000000 0.8964098 0.82347262 0.5583868 0.73927252 0.54501412 0.6300895

thigh 0.89640979 1.0000000 0.79917030 0.5397971 0.76147745 0.56684218 0.5586848

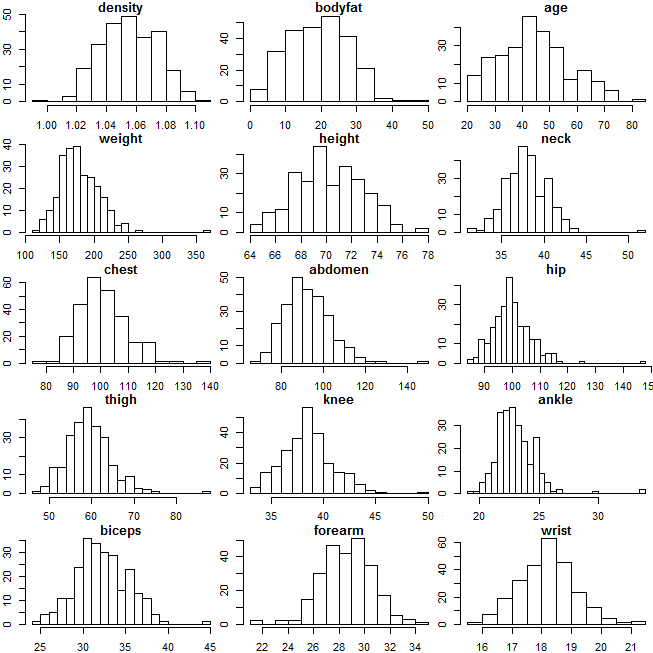
knee 0.82347262 0.7991703 1.00000000 0.6116082 0.67870883 0.55589819 0.6645073

ankle 0.55838682 0.5397971 0.61160820 1.0000000 0.48485454 0.41904999 0.5661946

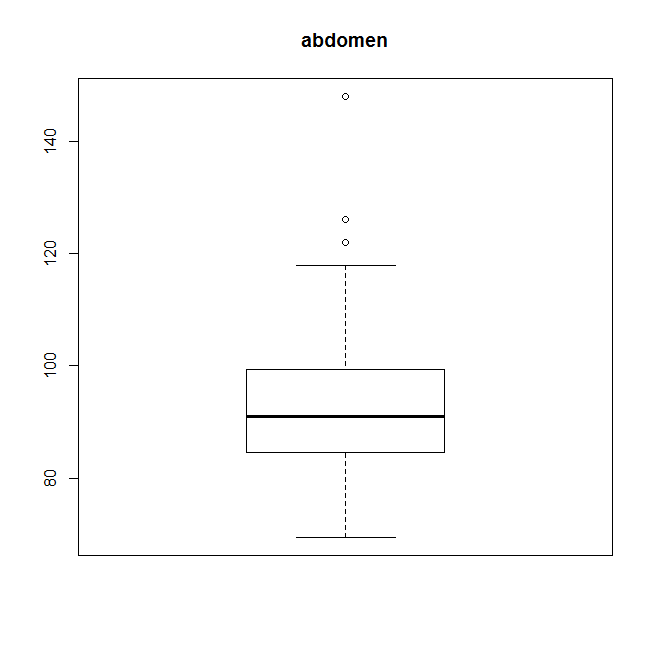
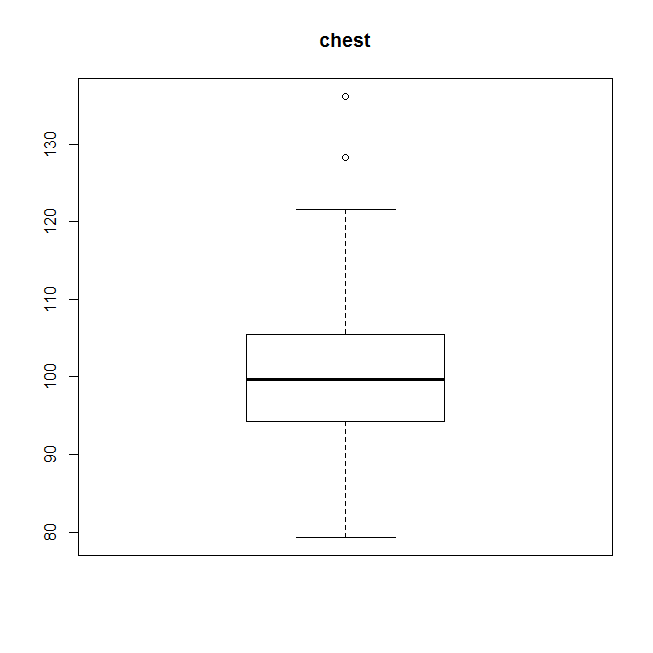
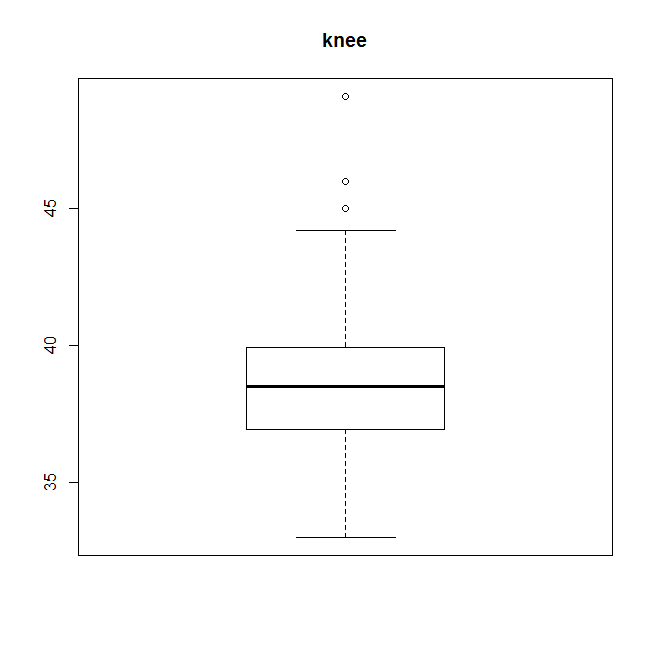
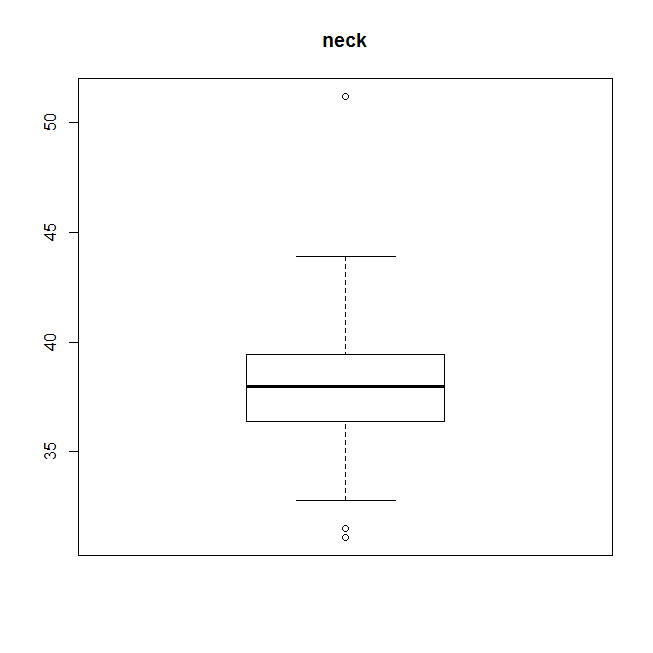
biceps 0.73927252 0.7614774 0.67870883 0.4848545 1.00000000 0.67825513 0.6321264

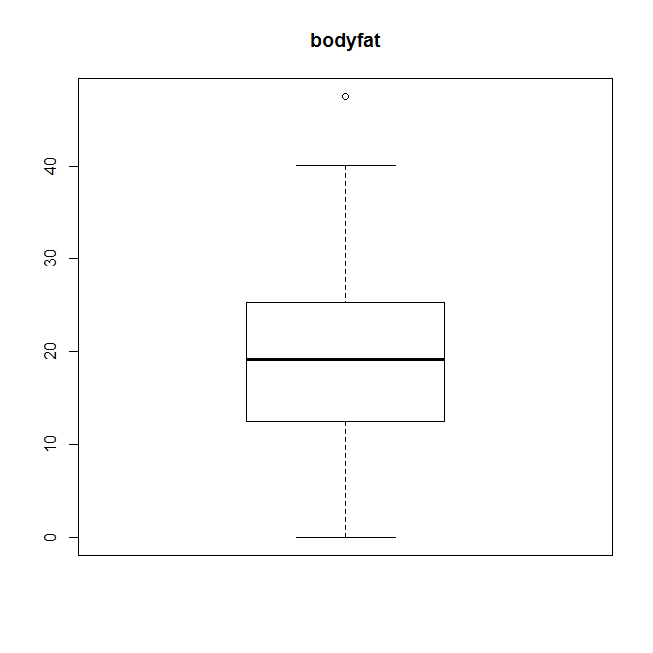
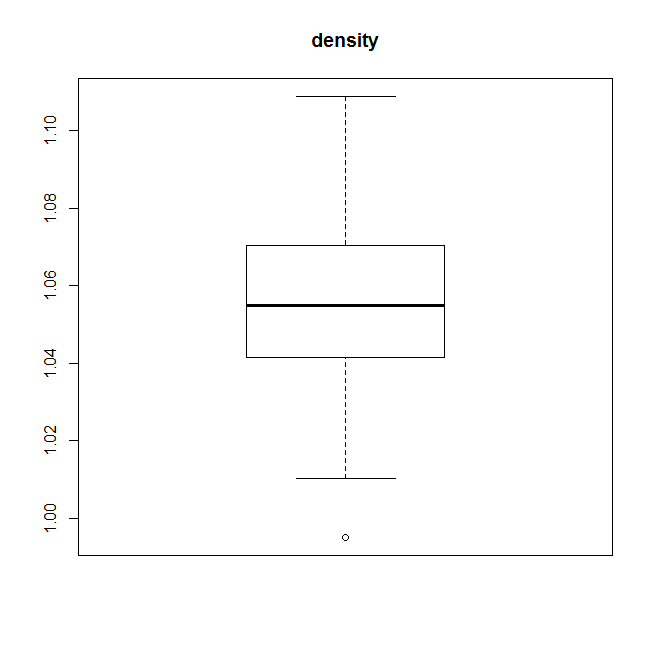
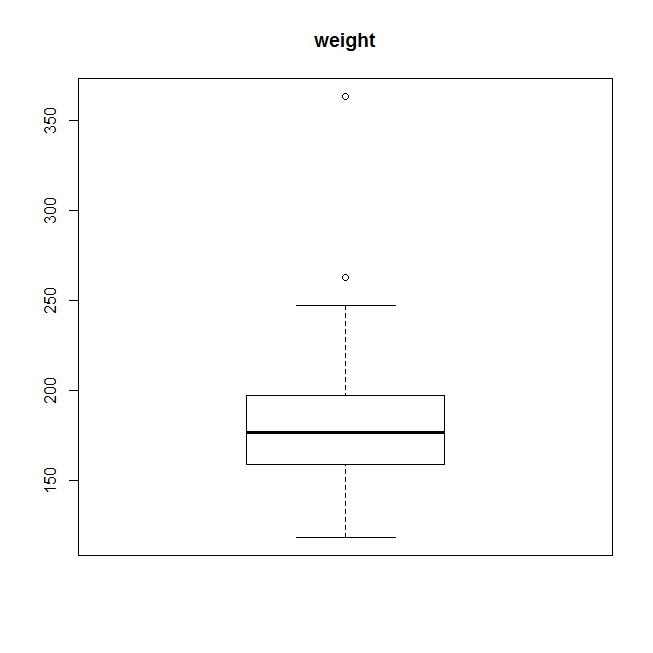
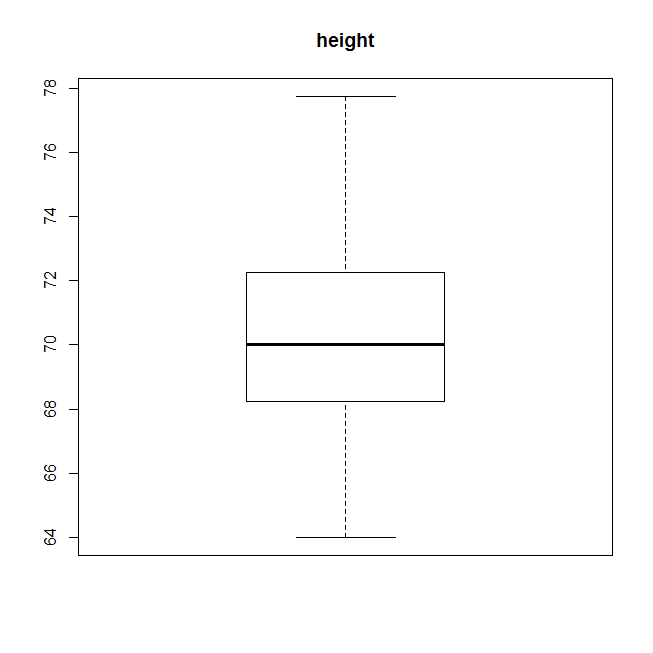
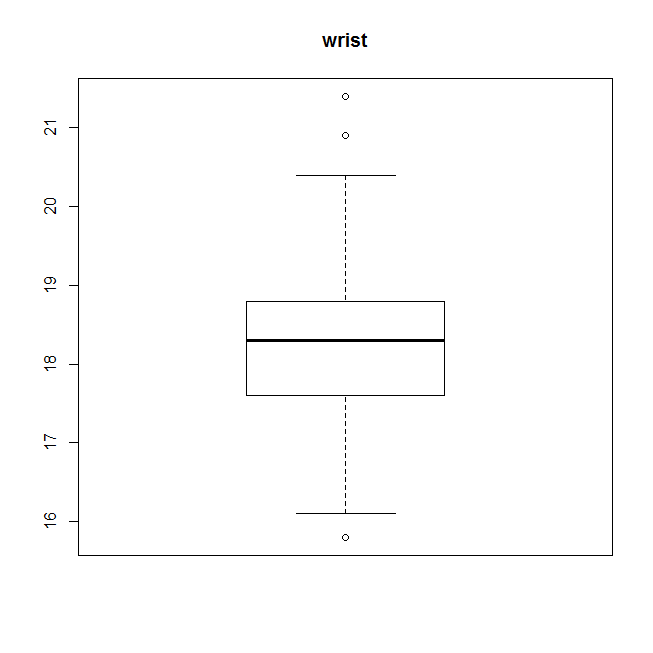
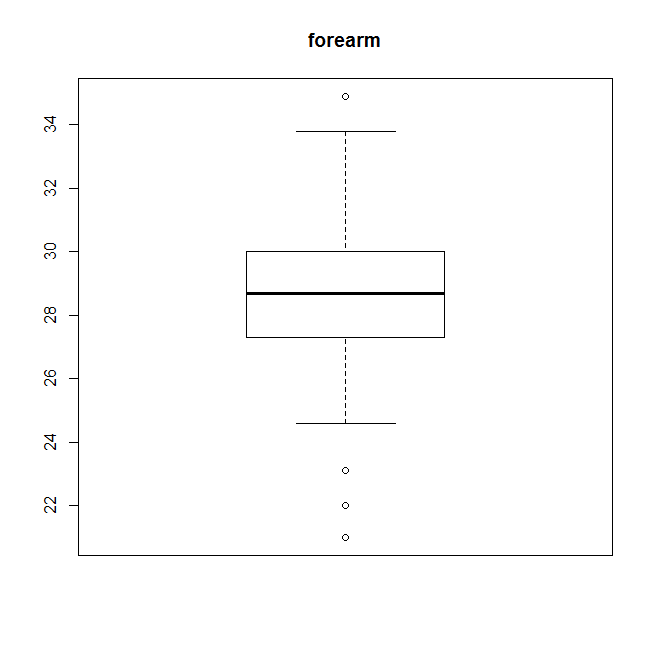
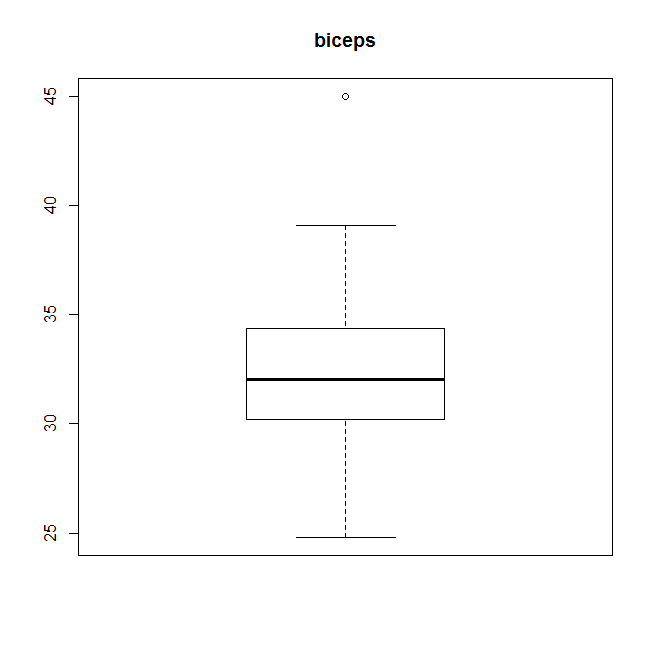
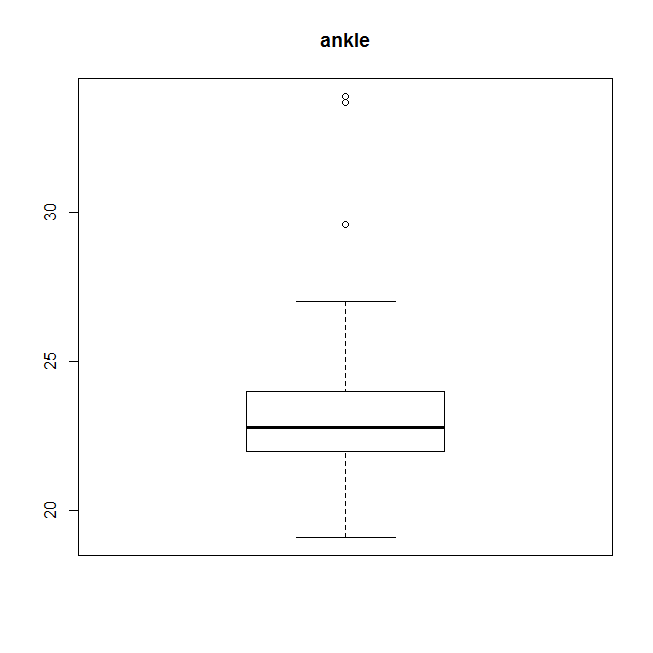
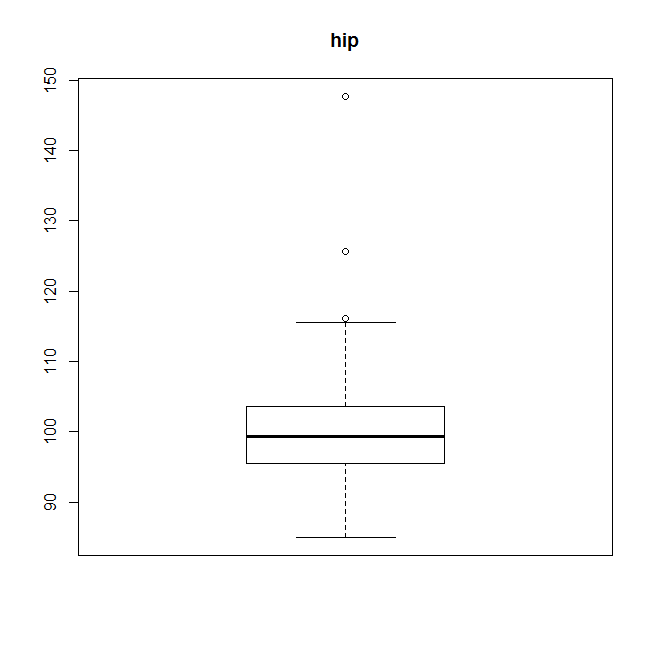
forearm 0.54501412 0.5668422 0.55589819 0.4190500 0.67825513 1.00000000 0.5855883

wrist 0.63008954 0.5586848 0.66450729 0.5661946 0.63212642 0.58558825 1.0000000

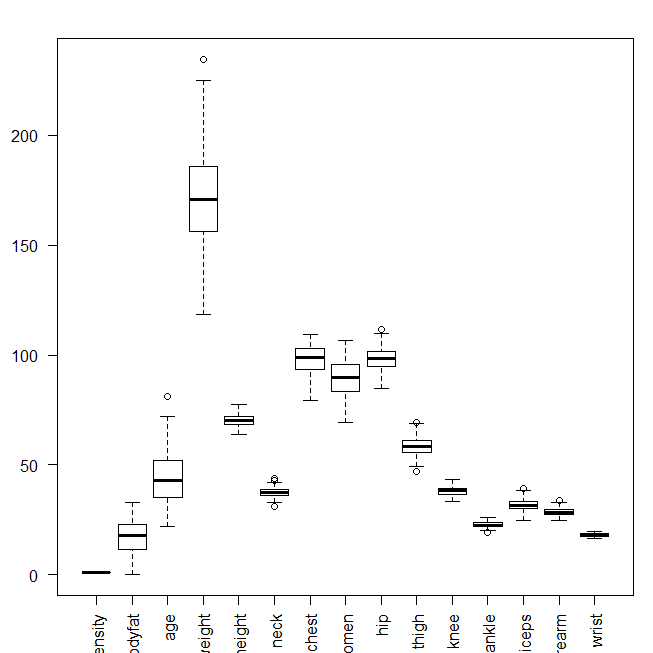


Individual Boxplots with the outliers





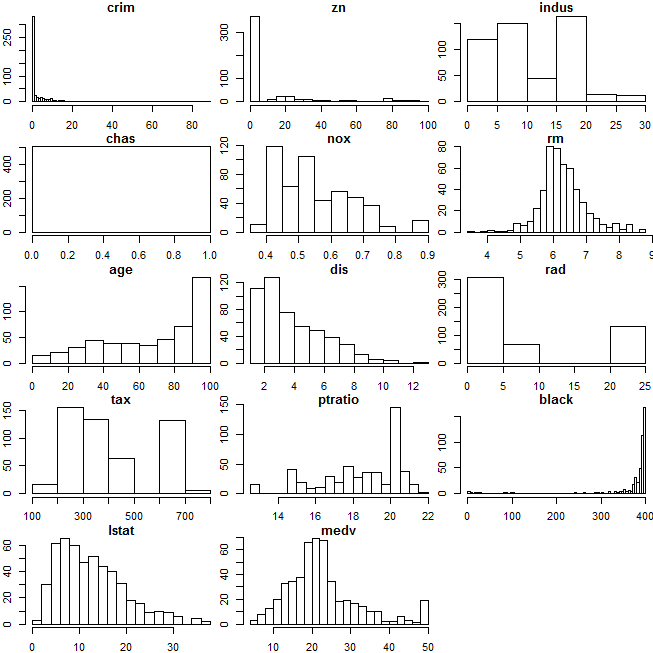
Boxplots after sub setting the data, the number of outliers have been reduced drastically.



**2) Question 3**

The Boston housing dataset was accessed through the ElemStatLearn Package.

1. Histograms of different variables are plotted to visualize the dataset.



Binary incidence matrix – The transactions class represents the data used for mining item sets or rules. The chas variable which represents the Charles River Dummy Variable was made to NULL because it is a dummy variable and there is no correlation to the other variables being used. Then the range of each variable was checked to get appropriate grouping categories. The ordered variables are dealt with first by plotting the histogram of the variable and checking the distribution.

First the “age” variable indicates proportion of owner-occupied units built prior to 1940, is categorized using the grouping where 0-25 is young, 25-45 is middle-aged, 45-65 is senior and 65-101 is elderly.

The “crim” variable indicates the per capita crime rate, which is categorized using the grouping where 0-10 is very low crime rate, 10-30 is low crime rate, 30-50 is medium crime rate, 50-90 is high crime rate. The histogram plotted shows the 0-10 region has the highest frequency and then the frequency reduces. High crime rate region which lies between 50-90 has the least frequency.

The “zn” variable indicates proportion of residential land zoned for lots over 25,000 sq.ft., which is categorized using the grouping where -5 to 20 is very low, 20-40 is low, 40-80 is medium and 80-101 is high. Here a negative value is taken to ensure there are no NA in the data and the minimum value 0 is taken onto account. The zn variable has the mean of 11.36 and the majority of its frequency lies in the very low region.

The “indus” variable which indicates proportion of non-retail business acres per town, is categorized using the grouping criteria where 0-7 is very low, 7-12 is low, 12-20 is medium and 20-28 is high proportion. It has a more or less an equal distribution in the very-low and low region and then in the very high region is has a high frequency.

The “nox” variable which indicates nitrogen oxides concentration has a very low range and hence the grouping is simple where 0.1-0.3 is very low concentration, 0.3-0.5 is low concentration, 0.5-0.8 is medium concentration, 0.8-1.0 is high concentration.

The “rm” variable indicates average number of rooms per dwelling, is categorized where 2-4 is a very low number, 4-6 is is a low number, 6-8 is a medium number, 8-10 is a high number.

The “dis” indicates weighted mean of distances to five Boston employment centres, which is categorized using the grouping criteria 0-4 is very low, 4-6 is low , 6-8 is medium, 8-13 is high.

The “rad” indicates index of accessibility to radial highways, which is categorized using the grouping criteria 0-5 is very low,5-12 is low, 12-18 is medium, 18-25 is high.

The “tax” indicates full-value property-tax rate per \$10,000, which is categorized using the grouping criteria 180-300 is very low tax rate, 300-500 is low tax rate, 500-600 is a medium tax rate, 600-750 is a high tax rate.

The “ptratio” indicates pupil-teacher ratio by town, which is categorized using the grouping criteria 12-16 is very low tax rate,16-18 is a low tax rate, 18-20 is a medium tax rate and 20-23 is a high tax rate.

The “black” indicates *1000(Bk - 0.63)^2* where *Bk* is the proportion of blacks by town, is categorized using the grouping criteria 0-100 is a very low proportion, 100-200 is a low proportion, 200-300 is a medium proportion, 300-400 is a high proportion.

The “lstat” indicates lower status of the population (percent), is categorized using the grouping criteria 0-10 is a very low percent, 10-20 is low percent, 20-30 is medium percent and 30-40 is a high percent.

The “medv” indicates median value of owner-occupied homes in \$1000s, which is categorized using the grouping criteria 0-10 is very low, 10-25 is low, 25-40 is medium, 40-60 is high.

**OUTPUT:**

> data1 <- as(data, "transactions")

> summary(data1)

transactions as itemMatrix in sparse format with

506 rows (elements/itemsets/transactions) and

49 columns (items) and a density of 0.2653061

most frequent items:

crim=very-low crime black=high zn=very-low medv=low nox=low

452 452 405 358 341

(Other)

4570

element (itemset/transaction) length distribution:

sizes

13

506

Min. 1st Qu. Median Mean 3rd Qu. Max.

13 13 13 13 13 13

includes extended item information - examples:

labels variables levels

1 crim=very-low crime crim very-low crime

2 crim=low crime crim low crime

3 crim=medium crime crim medium crime

includes extended transaction information - examples:

transactionID

1 1

2 2

3 3

1. The data was visualized using the itemFrequencyPlot in the “arules” package. itemFrequency() calculates the frequency for each item in an itemMatrix (Figure is on page -16). Conceptually, the item frequencies are the column sums of the binary matrix. The apriori algorithm is an algorithm for frequent item set mining and [association rule learning](https://en.wikipedia.org/wiki/Association_rule_learning) over transactional [databases](https://en.wikipedia.org/wiki/Databases). It proceeds by identifying the frequent individual items in the database and extending them to larger and larger item sets as long as those item sets appear sufficiently often in the database. The frequent item sets determined by Apriori can be used to determine [association rules](https://en.wikipedia.org/wiki/Association_rules).

The result of mining transaction data in arules are associations. Associations are sets of objects describing the relationship between some items which have assigned values for different measures of quality. The support which is a measure of significance is set to 0.001, the confidence which is a measure of interestingness is set to 0.6

The result of the mining algorithm is a set of 4399583 rules. The number of rules, show the most frequent items contained in the left-hand-side and the right-hand-side and their respective length distributions and summary statistics for the quality measures returned by the mining algorithm.

**OUTPUT:**

> rules <- apriori(data1, parameter = list(support = 0.001, confidence = 0.6))

Apriori

Parameter specification:

confidence minval smax arem aval originalSupport support minlen maxlen target ext

0.6 0.1 1 none FALSE TRUE 0.001 1 10 rules FALSE

Algorithmic control:

filter tree heap memopt load sort verbose

0.1 TRUE TRUE FALSE TRUE 2 TRUE

Absolute minimum support count: 0

set item appearances ...[0 item(s)] done [0.00s].

set transactions ...[49 item(s), 506 transaction(s)] done [0.00s].

sorting and recoding items ... [49 item(s)] done [0.00s].

creating transaction tree ... done [0.00s].

checking subsets of size 1 2 3 4 5 6 7 8 9 10 done [1.07s].

writing ... [4399583 rule(s)] done [3.21s].

creating S4 object ... done [7.78s].

> summary(rules)

set of 4399583 rules

rule length distribution (lhs + rhs):sizes

1 2 3 4 5 6 7 8 9 10

9 399 6764 50898 214334 562379 974797 1149855 930481 509667

Min. 1st Qu. Median Mean 3rd Qu. Max.

1.000 7.000 8.000 7.765 9.000 10.000

summary of quality measures:

support confidence lift

Min. :0.001976 Min. :0.6000 Min. : 0.6717

1st Qu.:0.001976 1st Qu.:1.0000 1st Qu.: 1.2494

Median :0.003953 Median :1.0000 Median : 1.6429

Mean :0.008572 Mean :0.9586 Mean : 2.8635

3rd Qu.:0.007905 3rd Qu.:1.0000 3rd Qu.: 2.8111

Max. :0.893281 Max. :1.0000 Max. :253.0000

mining info:

data ntransactions support confidence

data1 506 0.001 0.6

1. As typical for association rule mining, the number of rules found is huge. To analyze these rules, subset() of the crim is used to produce a separate subset of rules for the crim item which resulted from the variable income in the right-hand-side of the rule. Also the lift measure should exceed 1.2.

To check for a low crime rate area which is close to the city the inspect function by confidence is used.

The crim subset has a set a 23211 rules, which is again subset to check with very low dist i.e, the dis variable which has a set of 11256 rules.

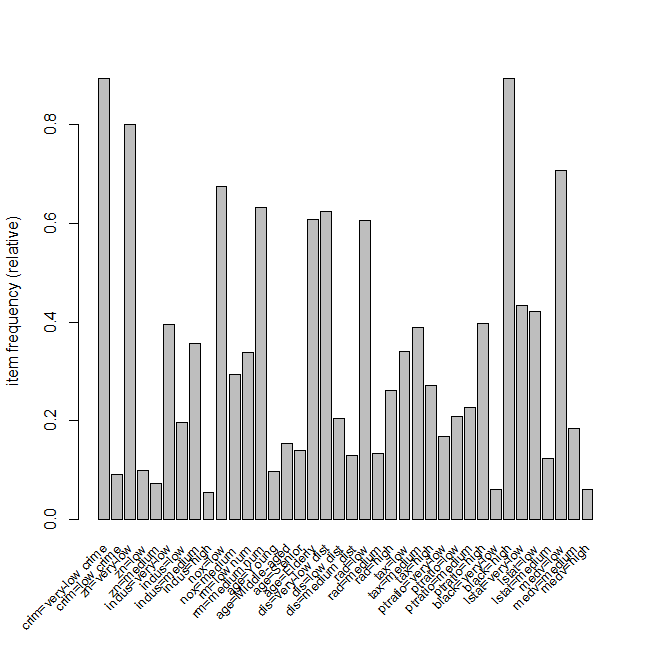
The crim subset has a lift of 11 and the crimdist has a lift of 11 too which shows a strong association.

From the first subset of rules we see that, for a low crime rate area the average number of rooms per dwelling must be very low, the black must be low, lstat should be high, the nox should be low.

From the second subset of rules for a low crime rate and very low dist , the rm is be very low, and black percent must be low, the lstat is high, rad is high, medv is medium.

So based on these parameters, the student looking for an area with low crime rate and closer to the city, must stay where the proportion of blacks is low, there is low number of rooms per dwelling, median value of owner-occupied homes is medium range, percent of lower status people is high.

|  |
| --- |
| OUTPUT :  > rulescrimSmall <- subset(rules, subset = rhs %in% "crim=low crime" & lift > 1.2)  > rulescrimSmall  set of 23211 rules  > inspect(head(sort(rulescrimSmall, by = "confidence"), n = 10))  lhs rhs support confidence lift  409 {rm=very-low num,black=low} => {crim=low crime} 0.001976285 1 11  429 {rm=very-low num,lstat=low} => {crim=low crime} 0.001976285 1 11  433 {rm=very-low num,medv=low} => {crim=low crime} 0.001976285 1 11  885 {black=low,medv=medium} => {crim=low crime} 0.001976285 1 11  904 {nox=low,black=low} => {crim=low crime} 0.001976285 1 11  1087 {black=medium,lstat=high} => {crim=low crime} 0.001976285 1 11  1793 {black=medium,medv=very-low} => {crim=low crime} 0.001976285 1 11  2803 {age=Senior,black=very-low} => {crim=low crime} 0.001976285 1 11  7174 {rm=very-low num,rad=high,black=low} => {crim=low crime} 0.001976285 1 11  7177 {rm=very-low num,tax=high,black=low} => {crim=low crime} 0.001976285 1 11 |
| |  | | --- | | > rulescrimdisSmall <- subset(rulescrimSmall, subset = lhs %in% "dis=very-low dist" & lift > 1.2)  > rulescrimdisSmall  set of 11256 rules  > inspect(head(sort(rulescrimdisSmall, by = "confidence"), n = 10))  lhs rhs support confidence lift  7195 {rm=very-low num,dis=very-low dist,black=low} => {crim=low crime} 0.001976285 1 11  7395 {rm=very-low num,dis=very-low dist,lstat=low} => {crim=low crime} 0.001976285 1 11  7411 {rm=very-low num,dis=very-low dist,medv=low} => {crim=low crime} 0.001976285 1 11  10414 {dis=very-low dist,black=low,medv=medium} => {crim=low crime} 0.001976285 1 11  10594 {nox=low,dis=very-low dist,black=low} => {crim=low crime} 0.001976285 1 11  11837 {dis=very-low dist,black=medium,lstat=high} => {crim=low crime} 0.001976285 1 11  12447 {nox=low,dis=very-low dist,lstat=high} => {crim=low crime} 0.001976285 1 11  17377 {dis=very-low dist,black=medium,medv=very-low} => {crim=low crime} 0.001976285 1 11  25046 {age=Senior,dis=very-low dist,black=very-low} => {crim=low crime} 0.001976285 1 11  58097 {rm=very-low num,dis=very-low dist,rad=high,black=low}=> {crim=low crime} 0.001976285 1 11 | |
| 1. d) A family is moving to the area, and has made schooling a priority. Schools with low pupil-teacher ratios are required. A subset 2. with a low ptratio is formed and a set of association rules is analyzed. A set of 52320 rules was found. The lift value is 4.81 and 3. confidence is 1. For the ptratio to be low , the dis is low, lstat is high, zn is medium, tax rate is low, rad is medium and medv is 4. medium. 5. From these parameters, a student looking for a school with low pupil-teacher ratio should have more distance to the Boston 6. Employment centers i.e. away from the main city, higher percent of the lower status people the accessibility to highways is 7. neither too far nor too near and the tax rate is low.   OUTPUT :  > rulesptratioSmall <- subset(rules, subset = rhs %in% "ptratio=low" & lift>1.2)  > rulesptratioSmall  set of 52320 rules  > inspect(head(sort(rulesptratioSmall, by = "confidence"), n = 10))  lhs rhs support confidence lift  1155 {dis=low dist,lstat=high} => {ptratio=low} 0.001976285 1 4.819048  1171 {tax=low,lstat=high} => {ptratio=low} 0.001976285 1 4.819048  1173 {indus=very-low,lstat=high} => {ptratio=low} 0.001976285 1 4.819048  1444 {rm=high num,medv=medium} => {ptratio=low} 0.003952569 1 4.819048  3271 {zn=medium,rad=medium} => {ptratio=low} 0.005928854 1 4.819048  4064 {indus=very-low,lstat=medium} => {ptratio=low} 0.001976285 1 4.819048  12467 {rm=low num,dis=low dist,lstat=high} => {ptratio=low} 0.001976285 1 4.819048  12470 {dis=low dist,tax=low,lstat=high} => {ptratio=low} 0.001976285 1 4.819048  12473 {indus=very-low,dis=low dist,lstat=high} => {ptratio=low} 0.001976285 1 4.819048  12476 {dis=low dist,rad=low,lstat=high} => {ptratio=low} 0.001976285 1 4.819048  ItemFrequency Plot |



1. Question 5

Consider the expression for

Then

If

Then the above equation can be written as:

Vectors is defined with the components given by

The above equation shows that

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*THE END \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\* \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*