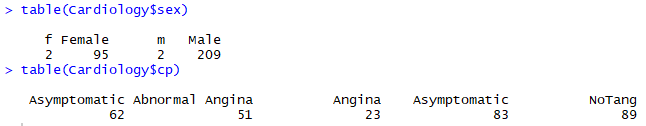
# Enterprise Database Technologies

# CA1

## Getting to know the Dataset using R

Part 1:

After receiving the dataset, I ran a few different commands to initially examine my dataset such as Summary and head. I then ran the table to command to check the columns in my dataset. By running the table command, I was able to get a more in depth look at the values being returned for each column and would be able to see any discrepancies in the data. This resulted in me finding out that two columns both had issues with data; the “Sex” and the “Cp” Column as shown below.



As you can see there are currently four values for “Sex”, when according to the documentation there should be only 2, Male and Female. The same can be seen in “CP” where Asymptomatic can be seen appearing twice. This is due to a white space in front of Asymptomatic meaning it is read as a separate variable. To more accurately evaluate the data, I decided to alter the values so that the extra values in “Sex” and “Cp” were moved back to where they were intended be.

Next, I moved on to find the percentage of missing values within the data. I ran a command on the dataset to query how many values were Na in the dataset. This returned me a result of 7 Na’s which gave me a percentage of 0.1517. I then ran the summary command on the dataset to quickly evaluate which columns had Na values. This showed me that the Cholesterol, Restecg and Class all had missing values and would need to have values imputed later.

Next, I found the max, min, mean, mode, median and the Standard deviation of the data to see if any information could be gained. The standard deviation was used to tell me how dispersed the data would be. A low standard deviation indicated it was closer to the mean and less dispersed, while a higher standard deviation indicated it the data was more dispersed.

The max and min of the data wasn’t useful in this case I wasn’t comparing to other values or reducing any of the vectors length. The same could be said with the mean, mode and median. On its own this data didn’t really give me great insight in how to evaluate the Cardiology dataset.

Next up I tested the type of distribution the data seemed to follow. The first test I ran was the Anderson Darling test, which tested for normality. If the resulting p-value was closer to one the more likely it would be normally distributed. However, I got extremely low values for all numeric attributes as shown in the numeric table in the Appendix. A limitation with the Anderson Darling test is that does not do well with large datasets, so I ran the Shapiro Wilks test to cross reference this outcome. I was again returned values that said the data was not normally distributed. Still not satisfied I researched further and found an explanation on stackoverflow(Fellows, 2018) that said these types of tests should **not** be used due to the fact they are null hypothesis tests **against** the assumption of normality. He goes onto say:

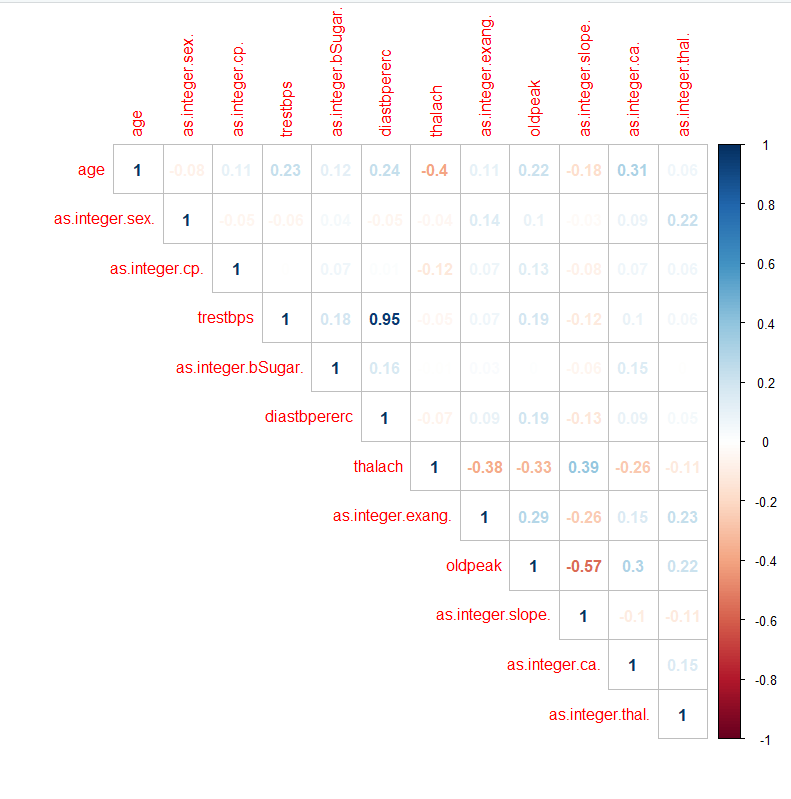
“When the sample size is small, even big departures from normality are not detected, and when your sample size is large, even the smallest deviation from normality will lead to a rejected null.”

So, my next step was to plot the quantile graphs along with histograms to determine normality in each of the numeric attributes which I believe gave me more accurate results. These graphs gave me 3 normally distributed and 3 not normally distributed graphs.

Based off the values I stored in the values table I had one symmetric value, 2 negatively skewed values and 3 positively skewed values. The most positively skewed value being oldpeak.



To calculate the level of correlation of the values I used corr. This would give me a large list of values comparing the correlation between each attribute. However, this was very difficult to read so I plot this data on a graph using corrplot as seen in below.



The more intense the colour is in the graph the more correlated it is. Blue indicates a positive correlation while red shows a negative correlation. It’s clear from the graph that diastbpexerc and trestbps have a strong correlation. I had to remove cholesterol and restecg from the graph due the fact they had na values and their correlation coefficients could not be calculated. At the moment no action should be taken except imputing missing values. This is due to the fact I am still processing the data.

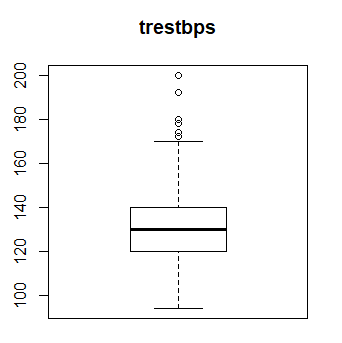
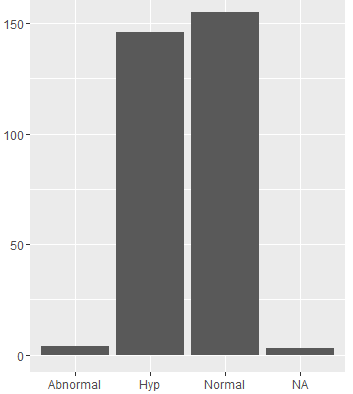
Next, I constructed histograms of the numerical data with overlays based on the target variable class. I did this by creating a function and running the attributes through this function. There was no discerning relationship between the cholesterol attribute and class. They both followed the same pattern with there being higher values of Healthy people compared to sick. For age there was a noticeable trend that after 50 there was an increase in sick people compared to those who were not sick. With diastbpererc I noticed that after a blood pressure of 90 you were more than likely to be sick than healthy. With oldpeak there was a clear increase in sick people as soon as the value went over 2.4. Another relationship could be seen with thalach and class as if you had a max heart rate above 145 then you were much more likely to be healthy. There was no real noticeable relationship with trestbps as class followed the same pattern for both variables.

I would expect Age to feature heavily in a machine learning algorithm due to the fact there is large noticeable change in sick and healthy people after a certain age is reached. The same could also be said for both oldpeak and thalach as they also showed significant changes after a certain point in their respective data.

After finding out relationships with the numeric data I then moved onto the Categorical data which I displayed using bar charts. Again, I created another function to display the barcharts. I started with “cp” and noticed straight away that one of the values dwarfed every other value in terms of sickness compared to health making it a prime candidate for a machine learning classification model. I felt that fasting blood sugar would have little significance in a machine learning model due the fact they didn’t show any sort of relationship. “exang” is another value that would be significant due to the difference in results between false and true; with true being more likely to be sick and false much more likely to be healthy. “slope”,”ca” and “thal” were more values which could be significant as one of the values was quite clearly healthy and the others were all sick. Also, being female gave a stronger chance of being healthy meaning that it also plays an important part in a machince learning classification model. I believe “restecg” wouldn’t have a significant impact due the fact the columns seemed evenly balanced between sick and healthy and that they were for the majority in one column.

I then moved onto checking for outliers. For the numerical values I used a boxplot to check for outliers. The boxplot shows outliers by displaying circles above the maximum and below the minimum values. The max and min in the boxplot exclude the outliers. The median is the dark black line and either side of the median is the upper and lower quartile.

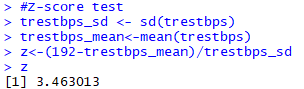
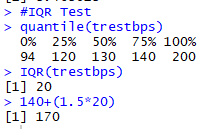
For the categorical data I used a bar chart excluding the class overlay in it. The bar chart indicates outliers based on the context. For example for “restecg”, abnormal would be classified as an outlier as there is so few people have an abnormal “restecg” in contrast to the other values excluding NA.

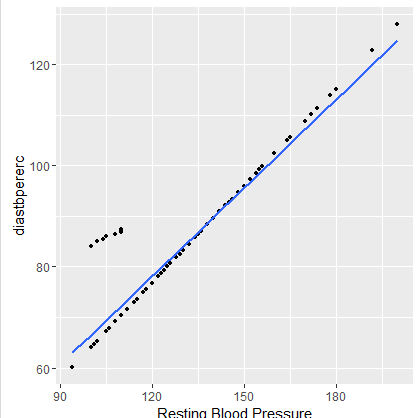
 

One statistical method I can use to confirm this is the Interquartile Range. If the Values are lower than Quartile1 – 1.5 \* (Interquartile Range) or higher than Quartile3 + 1.5 x (Interquartile Range).

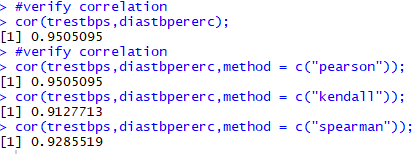
The second method is Z-score standardisation. This is calculated by subtracting the mean from the target number and dividing it by the standard deviation. If the values are either less then -3 or greater then 3.

Taking the second highest value 192 from “trestbps”, I ran this trough both tests and on both occasions, it was proved through to be an outlier.

I then began my search for correlated values. I created a 2D scatterplot using a function for each of the numeric values. Only one of the pairs returned correlated. 

This was the pair of “diastbpexerc” and “trestbps”. This can be verified using the corr function as shown below along with three different methods for calculating correlation; Pearson, Kendall and Spearmen. I chose the extra methods for more validation and the other methods backed up my original graph with none of the results going below .90.



Part 6: Cleaning dataset

First up was to discretise a numeric predictor variable using first “equal width binning” then afterwards k-means clustering. The variable I chose was age. I chose age because I wanted to see how the different age groups affected health performance based off my initial readings from the histograms earlier on. When choosing between equal width binning and K-means clustering its important to note that if the variable has outliers then the width of the bin can be affected by the presence of outliers. Whereas k-means clustering uses a clustering algorithm to calculate the optimal partitioning, therefore I believe the optimal solution is to move forward K-means Clustering.

Part7:

The numeric value I chose for transformation was the “oldpeak” variable is it was the most skewed. When running the 4 different tests trying to change the data I ran into issues. The first issue I had was that z-score returned the exact same skewness so that method became redundant. The next problem was that both natural log and Inverse square root transformation both returned NaN values due to the infinity value being reached for some the values in the vector. This left me with square root transformation. After running “oldpeak” through square root transformation it normalized the data and took away the positively skewed value for me. I then changed my original “oldpeak” data to the new normalized data for classification later.

Part 8:

The categorical variable I chose for the classification model was “restecg”. I chose this value because it had the most missing values in my dataset. I then filled the other missing values such as “cholesterol” with the median and “class” with the mode. I filled “class” because although it was the original target variable in the dataset it became a predictor variable when I chose “restecg” as the variable to be imputed.

The model I chose was the missForest algorithm. This algorithm is an implementation of the random forest algorithm. It is a non-parametric imputation method applicable to various variable types. This means that it does not make explicit assumptions about functional form of *f*(any arbitrary function). Instead, it tries to estimate *f* such that it can be as close to the data points without seeming impractical. It works by setting up a random forest model for each variable. Then it predicts missing values in the variable with help of observed values. The model provides a high level control of the imputation process. The model returns OOB (out of bag) imputation error estimate. It allows options to return OOB separately rather than for the whole data matrix. This allows us to look more closely as to how accurately the model has imputed values for each variable. (ANALYTICS VIDHYA CONTENT TEAM , 2018)

According to the random forest official page it doesn’t matter how many trees are included when running the algorithm it will never over fit. (Breiman & Adele, 2018) So I tried the default value, 20 and 40 for the number of trees. I found that the values imputed for the default and 20 trees returned the same values “hyp”,”Normal”,”Normal”. However once I increased the trees to size of 40 it returned 3 normal values. This would indicate…

Part 9:

# References

ANALYTICS VIDHYA CONTENT TEAM . (2018, March 13). *Tutorial on 5 Powerful R Packages used for imputing missing values*. Retrieved from analytics vidhya: https://www.analyticsvidhya.com/blog/2016/03/tutorial-powerful-packages-imputing-missing-values/

Breiman, L., & Adele, C. (2018, March 13). *Random Forests*. Retrieved from stat.berkeley: https://www.stat.berkeley.edu/~breiman/RandomForests/cc\_home.htm#remarks

Fellows, I. (2018, March 12). *Seeing if data is normally distributed in R*. Retrieved from StackOverflow: https://stackoverflow.com/questions/7781798/seeing-if-data-is-normally-distributed-in-r/7788452#7788452