Data Pre-processing

Pima Indians Diabetes Dataset

Table of Contents

1) Introduction

Diabetes mellitus – commonly known as diabetes – is a lifelong metabolic disorder characterized by abnormally high levels of blood glucose, and inefficient insulin action. The disease often leads to significant disability, including kidney failure, blindness, limb amputation, and premature death. It is one of the major noncommunicable diseases which have great impact on human life today.

The Pima Indians have the highest reported prevalence and incidence of diabetes of any population in the world. They are a group of Native Americans who mostly reside in Phoenix, Arizona, USA. They have minimal European admixture, and their diabetes appears to be exclusively type 2, with no evidence of the autoimmunity characteristics of type 1, even in very young subjects with an early onset of the disease. Diabetes in Pima Indians is also familial, and the degree of familiality is greater at younger ages of onset compared with older ages of onset (Baier and Hanson, 2004).

Diabetes was very rare among Native Americans until the middle part of the twentieth century. However, since World War II, high rates of the disease have been observed among many Native American tribes, as well as other diverse societies worldwide that have recently adapted to western culture. It has quickly become one of the most common serious diseases affecting these demographics (Venkat Narayan, 1997).

The dataset used in this report comes from the USA's National Institute of Diabetes and Digestive and Kidney Diseases. The purpose of this dataset is to determine whether or not a patient is diabetic, on the basis of certain diagnostic measures, including the patient's number of pregnancies, BMI levels, insulin levels, age, etc. The dataset comprises medical reports of 768 Pima Indian women, at least 21 years of age. The samples consists of 8 variables of the medical predictor and one of the two objective outcomes; namely whether the patient tested positive for diabetes or not.

Gathered data often contains anomalies, impossible data combinations, missing values, etc. Analysing any dataset before it has been screened for such errors can produce misleading results. Thus, this report aims to pre-process the "PimaIndiansDiabetes" dataset using RStudio, in preparation for further analysis. The data set can be found at (Leisch, n.d.)

2) Data Cleaning:

Data cleaning (or cleansing) is the first step of data pre-processing. It involves detecting incomplete, incorrect, inaccurate, or irrelevant entries in a dataset, then modifying, and/or removing said "dirty data". This process may also involve removing typographical errors or validating and correcting values against a list of known entities.

2.1 Data Summarisation:

The dataset contains 8 columns with continuous values, and 768 entries. The variables are:

- Pregnant: Number of times pregnant.
- Glucose: Plasma glucose concentration over 2 hours in an oral glucose tolerance test.
- Pressure: Diastolic blood pressure (mmHg).
- Triceps: Triceps skin fold thickness (mm).
- Insulin: 2-Hour serum insulin (mu U/ml).
- Mass: Body mass index (weight in kg/(height in m^2)).
- Pedigree: Diabetes pedigree function (a function which scores likelihood of diabetes based on family history).
- Age: Age in years.
- Diabetes: Outcome (neg if non-diabetic, pos if diabetic).

First, we will be using summarisation functions in R to explore the data and determine if it's clean.

```
> rm(list = ls(all = TRUE))
> library(mlbench)
> data("PimaIndiansDiabetes")
> summary(PimaIndiansDiabetes)
    pregnant         glucose          pressure         triceps     
 Min.   : 0.000   Min.   :  0.0   Min.   :  0.00   Min.   : 0.00  
 1st Qu.: 1.000   1st Qu.: 99.0   1st Qu.: 62.00   1st Qu.: 0.00  
 Median : 3.000   Median :117.0   Median : 72.00   Median :23.00  
 Mean   : 3.845   Mean   :120.9   Mean   : 69.11   Mean   :20.54  
 3rd Qu.: 6.000   3rd Qu.:140.2   3rd Qu.: 80.00   3rd Qu.:32.00  
 Max.   :17.000   Max.   :199.0   Max.   :122.00   Max.   :99.00  
    insulin           mass          pedigree           age        
 Min.   :  0.0   Min.   : 0.00   Min.   :0.0780   Min.   :21.00  
 1st Qu.:  0.0   1st Qu.:27.30   1st Qu.:0.2437   1st Qu.:24.00  
 Median : 30.5   Median :32.00   Median :0.3725   Median :29.00  
 Mean   : 79.8   Mean   :31.99   Mean   :0.4719   Mean   :33.24  
 3rd Qu.:127.2   3rd Qu.:36.60   3rd Qu.:0.6262   3rd Qu.:41.00  
 Max.   :846.0   Max.   :67.10   Max.   :2.4200   Max.   :81.00  
 diabetes
 neg:500
 pos:268
```

The summary table above gives a general overview of the data. Looking at every column individually, we can note that:

- The continuous variables "pregnant", "pedigree", and "age", seem to give normal results with no missing or illogical values.
- The remaining variables contain some impossible values. To take a closer look at these variables, we'll generate histograms for each one.

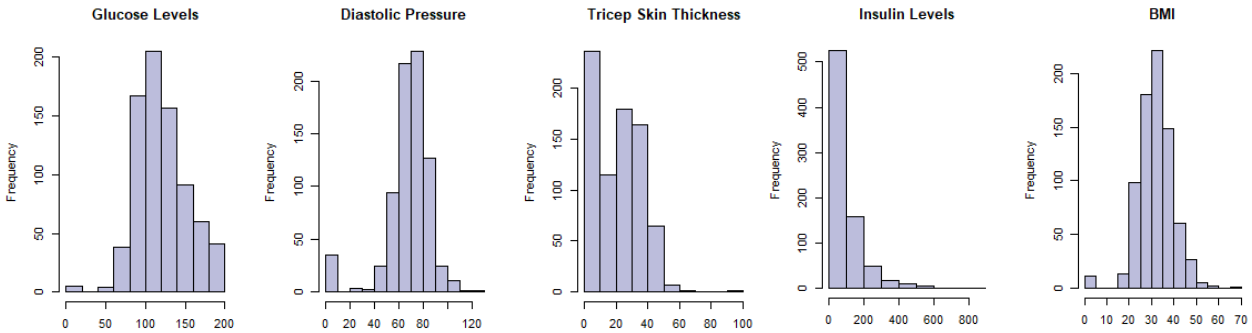

Figure 2.1 – histograms of continuous diagnostic variables

From figure 2.1, we can see that all the variables have some odd entries which are impossible in real life; a person cannot have skin thickness of 0 mm, or blood pressure below 40 mmHg, etcetera.

Thus, the data does contains some unexpected values, and needs further cleaning before it can be analysed.

2.2 Outlier Detection:

There are many ways different ways of finding outliers in a dataset. For this section, we will use the "IQR-based approach" – it's a very commonly non-parametric approach used to determine whether the continuous columns in a dataset have any outliers, then identify which values they are.

The IQR is the interquartile range, given by the difference between the two quartiles 1 and 3. Once the IQR is found, treating outliers is then executed as follows:

- Multiply the interquartile range (IQR) by 1.5 (a constant used to discern outliers).
- Add 1.5 × (IQR) to the third quartile. Any number greater than this is a suspected outlier.
- Subtract 1.5 × (IQR) from the first quartile. Any number less than this is a suspected outlier.

We can use code in R to find the IQR for a specific continuous column in a dataset, and perform the operations above to determine where the outliers are.

We'll start with the first variable, "pregnant":

```
> iqr <- IQR(PimaIndiansDiabetes$pregnant) # IQR range
> Q1 <- quantile(PimaIndiansDiabetes$pregnant, 0.25)
> Q3 <- quantile(PimaIndiansDiabetes$pregnant, 0.75)
> as.numeric(c(Q1-1.5*iqr, Q3+1.5*iqr))
[1] -6.5 13.5
>
> summary(PimaIndiansDiabetes$pregnant)
   Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
  0.000   1.000   3.000   3.845   6.000  17.000
>
```

We can see from the code above that the IQR for this variable is [– 6.5, 13.5]. Thus, the number of pregnancies should fall within this interval. From the summary function, we see that the range in the dataset is [0, 17] – so we need to eliminate subjects with 13, 14, 15, 16, and 17 pregnancies.

Repeating the code over all the columns, we get the following IQRs:

- pregnant: [– 6.5, 13.5]
- glucose: [37.125, 202.125]
- pressure: [35, 107]
- triceps: [– 48, 80]
- insulin: [– 190.875, 318.125]
- mass: [13.35, 50.55]
- pedigree: [– 0.33, 1.20]
- age: [– 1.5, 66.5]

Anything that lies outside of these ranges is an outlier. We use the following code to find the ranges for all the remaining continuous columns:

```
> for (i in names(PimaIndiansDiabetes)) {
+   s <- sprintf("data filed %s range: [%s-%s]",
+                i, range(PimaIndiansDiabetes[,i])[1],
+                range(PimaIndiansDiabetes[,i])[2])
+   print(s)
+ }
[1] "data filed pregnant range: [0-17]"
[1] "data filed glucose range: [0-199]"
[1] "data filed pressure range: [0-122]"
[1] "data filed triceps range: [0-99]"
[1] "data filed insulin range: [0-846]"
[1] "data filed mass range: [0-67.1]"
[1] "data filed pedigree range: [0.078-2.42]"
[1] "data filed age range: [21-81]"
```

Analysing each of the IQRs, there are two things to observe:

- All the columns seem to contain outliers, however, the missing values discussed in section 2.1 happen to fall within the IQR range. So we will set them to "NA" along with any outliers that do not equal 0.
- The only outlier for the "age" variable is 81 – however, it is not abnormal to have someone who is 81 years old, and it is consistent with the rest of the results. From this summary table below, we can see that the person who is 81 years old has normal results for the other variables, none of them lying outside the IQR.

```
> PimaIndiansDiabetes[460,]
    pregnant glucose pressure triceps insulin mass pedigree age
460        9     134       74      33      60 25.9     0.46  81

diabetes neg
>
```

Hence, while the IQR rule generally holds, it does not apply to every case. An overall outlier analysis should be followed by examining each potential outlier in the context of the entire dataset, to ensure no correct or valuable data is lost.

After identifying the outliers and missing values, we can set them equal to "NA".

```
# Create new dataset to store cleaned data with no missing values or
outliers.
Pima_NA<-PimaIndiansDiabetes
```

```
View(Pima_NA)

# Change specific missing values or outlier ranges to NA.
class(Pima_NA$pregnant)
Pima_NA$pregnant[Pima_NA$pregnant>13.5] <- "NA"

class(Pima_NA$glucose)
Pima_NA$glucose[Pima_NA$glucose %in% c("0", "<37.125", ">202.125")]
<- "NA"

class(Pima_NA$pressure)
Pima_NA$pressure[Pima_NA$pressure %in% c("0", "<35", ">107")] <-
"NA"

class(Pima_NA$triceps)
Pima_NA$triceps[Pima_NA$triceps %in% c("0", ">80")] <- "NA"

class(Pima_NA$insulin)
Pima_NA$insulin[Pima_NA$insulin %in% c("0", ">318.125")] <- "NA"

class(Pima_NA$mass)
Pima_NA$mass[Pima_NA$mass %in% c("0", "<13.35", ">50.55")] <- "NA"

class(Pima_NA$pedigree)
Pima_NA$pedigree[Pima_NA$pedigree %in% c("<-0.33", ">1.20")] <- "NA"

# Convert back into a numeric type argument (real or integer).
for (i in 1:8) Pima_NA[,i]<-as.numeric(Pima_NA[,i])

# Summarise the new dataset.
summary(Pima_NA)
```

After removing the unwanted values, the summary function shows us the new dataset with cleaner values overall and the number of outliers set to NA.

```
> summary(Pima_NA)
    pregnant         glucose         pressure
 Min.   : 0.000   Min.   : 44.0   Min.   : 24.00
 1st Qu.: 1.000   1st Qu.: 99.0   1st Qu.: 64.00
 Median : 3.000   Median :117.0   Median : 72.00
 Mean   : 3.787   Mean   :121.7   Mean   : 72.41
 3rd Qu.: 6.000   3rd Qu.:141.0   3rd Qu.: 80.00
 Max.   :13.000   Max.   :199.0   Max.   :122.00
 NA's   :4        NA's   :5       NA's   :35
    triceps          insulin            mass
 Min.   : 7.00   Min.   : 14.00   Min.   :18.20
 1st Qu.:22.00   1st Qu.: 76.25   1st Qu.:27.50
 Median :29.00   Median :125.00   Median :32.30
 Mean   :29.15   Mean   :155.55   Mean   :32.46
 3rd Qu.:36.00   3rd Qu.:190.00   3rd Qu.:36.60
 Max.   :99.00   Max.   :846.00   Max.   :67.10
 NA's   :227     NA's   :374      NA's   :11
```

```
    pedigree           age         diabetes
 Min.   :0.0780   Min.   :21.00   neg:500
 1st Qu.:0.2437   1st Qu.:24.00   pos:268
 Median :0.3725   Median :29.00
 Mean   :0.4719   Mean   :33.24
 3rd Qu.:0.6262   3rd Qu.:41.00
 Max.   :2.4200   Max.   :81.00
```

2.3 Mean Imputation:

Mean imputation is a popular method used to deal with missing data. It works by replacing the missing observations in a dataset with the mean of the non-missing observations for that variable.

For PimaIndiansDiabetes, we will use mean imputation on all the values we set to NA in the previous section.

```
# Create new dataset for mean imputation.
Pima_mean_imp<- Pima_NA

# Mean imputation of missing values and outliers.
for (i in 1:7)
  Pima_mean_imp[is.na(Pima_NA[,i]), i]<- mean(Pima_NA[,i],
na.rm=TRUE)

    pregnant         glucose          pressure
 Min.   : 0.000   Min.   : 44.00   Min.   : 24.00
 1st Qu.: 1.000   1st Qu.: 99.75   1st Qu.: 64.00
 Median : 3.000   Median :117.00   Median : 72.20
 Mean   : 3.787   Mean   :121.69   Mean   : 72.41
 3rd Qu.: 6.000   3rd Qu.:140.25   3rd Qu.: 80.00
 Max.   :13.000   Max.   :199.00   Max.   :122.00
    triceps          insulin           mass
 Min.   : 7.00   Min.   : 14.0   Min.   :18.20
 1st Qu.:25.00   1st Qu.:121.5   1st Qu.:27.50
 Median :29.15   Median :155.5   Median :32.40
 Mean   :29.15   Mean   :155.5   Mean   :32.46
 3rd Qu.:32.00   3rd Qu.:155.5   3rd Qu.:36.60
 Max.   :99.00   Max.   :846.0   Max.   :67.10
    pedigree           age         diabetes
 Min.   :0.0780   Min.   :21.00   neg:500
 1st Qu.:0.2437   1st Qu.:24.00   pos:268
 Median :0.3725   Median :29.00
 Mean   :0.4719   Mean   :33.24
 3rd Qu.:0.6262   3rd Qu.:41.00
 Max.   :2.4200   Max.   :81.00
```

From the summary table of the imputed dataset, we can see the subtle changes in values after imputation.

3) Data Normalisation:

Normalisation is a step in data cleaning and pre-processing, where values that are measured on different scales are adjusted to one notionally common scale. Sometimes, quantile normalisation is used, to bring all probability distributions of adjusted values into alignment, making them identical in statistical properties. One of its advantages is increased consistency, as the information is stored in one place only, thus, reducing the possibility of inconsistent data.

3.1 Z-score Normalisation:

Z-score normalisation is a strategy of normalising data that can also be used as an outlier detection method. Most often, normalisation refers to the rescaling of continuous columns in a dataset to a range of [0, 1]. Using standardisation, we centre the continuous columns at mean 0 and standard deviation 1, making them take the form of a normal distribution.

The formula for Z-score normalisation is:

$$Z = \frac{x_i - \mu}{\sigma}$$

where:

$Z =$ standard score, $x_i =$ observed value, $\mu =$ mean, $\sigma =$ standard deviation.

The continuous variables of interest are transformed the Z-score. Then, it is often used in standardised testing, where the value is compared to a fixed prediction threshold, such as 2.5, 3, and 3.5.

For PimaIndiansDiabetes, we will normalise all continuous columns using the Z-score.

```
# Create new dataset to store the standardised data.
Pima_stand<- Pima_mean_imp
View(Pima_stand)

# Data normalisation using Z-score.
library(robustHD)
for (i in 1:8)
  Pima_stand[,i]<-robustHD::standardize(Pima_stand[,i])
```

The data is now normalised, we will use a statistical summary function in R to see the changes this process had on the mean and standard deviation in the continuous columns of the dataset.

Summary of both datasets to compare the changes.

```
> # Or, for specific columns only.
> describe(Pima_mean_imp[, c(1:8)])
         vars   n   mean    sd median trimmed   mad
pregnant    1 768   3.79  3.27   3.00    3.43  2.97
glucose     2 768 121.69 30.44 117.00  119.67 29.65
pressure    3 768  72.41 12.10  72.20   72.29 11.56
triceps     4 768  29.15  8.79  29.15   28.97  5.70
insulin     5 768 155.55 85.02 155.55  145.28  5.19
mass        6 768  32.46  6.88  32.40   32.11  6.82
pedigree    7 768   0.47  0.33   0.37    0.42  0.25
age         8 768  33.24 11.76  29.00   31.54 10.38
           min    max  range skew kurtosis   se
```

```
pregnant  0.00  13.00  13.00 0.82    -0.16 0.12
glucose  44.00 199.00 155.00 0.53    -0.27 1.10
pressure 24.00 122.00  98.00 0.14     1.07 0.44
triceps   7.00  99.00  92.00 0.82     5.35 0.32
insulin  14.00 846.00 832.00 3.01    15.03 3.07
mass     18.20  67.10  48.90 0.60     0.90 0.25
pedigree  0.08   2.42   2.34 1.91     5.53 0.01
age      21.00  81.00  60.00 1.13     0.62 0.42
> describe(Pima_stand[, c(1:8)])
         vars   n mean sd median trimmed  mad   min
pregnant    1 768    0  1  -0.24   -0.11 0.91 -1.16
glucose     2 768    0  1  -0.15   -0.07 0.97 -2.55
pressure    3 768    0  1  -0.02   -0.01 0.96 -4.00
triceps     4 768    0  1   0.00   -0.02 0.65 -2.52
insulin     5 768    0  1   0.00   -0.12 0.06 -1.66
mass        6 768    0  1  -0.01   -0.05 0.99 -2.07
pedigree    7 768    0  1  -0.30   -0.15 0.75 -1.19
age         8 768    0  1  -0.36   -0.14 0.88 -1.04
          max range skew kurtosis   se
pregnant 2.82  3.98 0.82    -0.16 0.04
glucose  2.54  5.09 0.53    -0.27 0.04
pressure 4.10  8.10 0.14     1.07 0.04
triceps  7.95 10.47 0.82     5.35 0.04
insulin  8.12  9.79 3.01    15.03 0.04
mass     5.04  7.11 0.60     0.90 0.04
pedigree 5.88  7.07 1.91     5.53 0.04
age      4.06  5.10 1.13     0.62 0.04
>
```

As we can see from the summary tables above, the standard deviation value (sd) and mean are now all equal to 1 and 0 respectively, as opposed to the varied values they had before normalisation. This means that the dataset is now following a standard normal distribution.

3.2 Normality Hypothesis Test:

In this section, we will find how likely or unlikely it is that we have obtained the data that we observed. This can be tested in many ways. Now that we have that the data follows a standard normal distribution, we can assume normality of the dataset. We define the null-hypothesis as; the data being normally distributed. We will use the Shapiro-Wilk test to generate a p-value that can be compared against an existing threshold.

p-value	*evidence against null-hypothesis*
$p > 0.10$	none
$0.05 < p < 0.10$	weak
$0.01 < p < 0.05$	strong
$0.001 < p < 0.01$	very strong
$P \leq 0.001$	very, very strong

The lower the p-value, the less likely it is for this data to be generated under the null-hypothesis – i.e. there is a very low chance of this dataset being produced under normal distribution. This means the null-hypothesis will be rejected. On the other hand, if the p-value is large compared to the above threshold, the null-hypothesis cannot be rejected, as that will be strong evidence that the data came from a normally distributed sample.

To check the null-hypothesis for PimaIndiansDiabetes, we will first use the Shapiro-Wilk test in R:

```
> # Runing the Shapiro-Wilk normality test on every column.
> shapiro.test(Pima_NA$pregnant)

     Shapiro-Wilk normality test

data:  Pima_NA$pregnant
W = 0.90583, p-value < 2.2e-16

> shapiro.test(Pima_NA$glucose)

     Shapiro-Wilk normality test

data:  Pima_NA$glucose
W = 0.96964, p-value = 1.72e-11

> shapiro.test(Pima_NA$pressure)

     Shapiro-Wilk normality test

data:  Pima_NA$pressure
W = 0.99031, p-value = 9.451e-05

> shapiro.test(Pima_NA$triceps)

     Shapiro-Wilk normality test

data:  Pima_NA$triceps
W = 0.968, p-value = 1.776e-09

> shapiro.test(Pima_NA$insulin)

     Shapiro-Wilk normality test

data:  Pima_NA$insulin
W = 0.8041, p-value < 2.2e-16

> shapiro.test(Pima_NA$mass)

     Shapiro-Wilk normality test

data:  Pima_NA$mass
W = 0.97955, p-value = 8.558e-09

> shapiro.test(Pima_NA$pedigree)
```

```
	Shapiro-Wilk normality test

data:  Pima_NA$pedigree
W = 0.83652, p-value < 2.2e-16

> shapiro.test(Pima_NA$age)

	Shapiro-Wilk normality test

data:  Pima_NA$age
W = 0.87477, p-value < 2.2e-16

>
```

As shown by the results of the operation above, all the p-values are extremely small, falling in the range $p \leq 0.001$ – thus, the normality hypothesis is not satisfied for any continuous columns in the dataset. We conclude that the data could not be generated from a normally distributed sample.

4) Data Transformation; Response Transformation:

Often, the assumptions of the modelling techniques are not satisfied. In these cases, a data transformation is performed as a way of dealing with the problems in the dataset. There are various methods of transformation, depending on the assumption of the modelling technique.

In the case of PimaIndiansDiabetes, the Shapiro-Wilk test showed that the data does not follow a normal distribution. One solution to this is to transform the dataset using a Box-Cox transformation.

The Box-Cox algorithm uses logs, and works by expanding the differences between smaller values, and reducing it between larger ones. This is because the slope of the logarithmic function is steeper when values are small but when they are larger. If we inflate the difference on one end of the spectrum while reducing the difference on the other, the result will be a symmetrical normal distribution. The formula for one-parameter Box-Cox transformation is defined as:

$$g_\lambda(y) = \begin{cases} \dfrac{y^\lambda - 1}{\lambda} & if \ \ \lambda \neq 0 \\ \log(y) & if \ \ \lambda = 0 \end{cases}$$

where the λ parameter is chosen by numerically maximising the log-likelihood function:

$$L(\lambda) = \frac{-n}{2} log\left(\frac{RSS_\lambda}{n}\right) + (\lambda - 1)\sum log(y_i)$$

then perform a Box-Cox transformation (Box and Cox, 1964).

First, we use R functions on two continuous columns in the dataset to find the value of λ:

```
> # Finding lambda.
> library(MASS)
> transf <- boxcox(Pima_NA$glucose ~ Pima_NA$pressure)
> lambda <- transf$x[which.max(transf$y)]
```

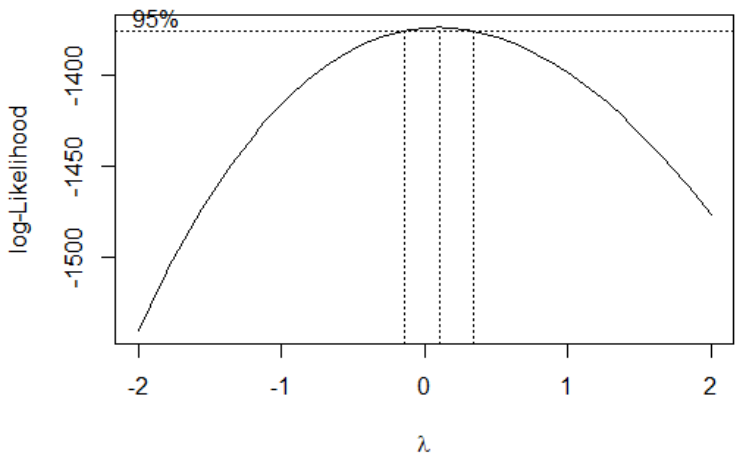

Figure 4.1 – λ curve generated in R

The code generated the value $\lambda = 0.101010101010101$

Now we can perform the Box-Cox transformation and use it to produce the Q-Q plot for the original and transformed values. For this we will be using the continuous columns "glucose" and "pressure."

```
> # Box-Cox transformation performed on dataset with no zeros or
negatives.
> library(MASS)
```

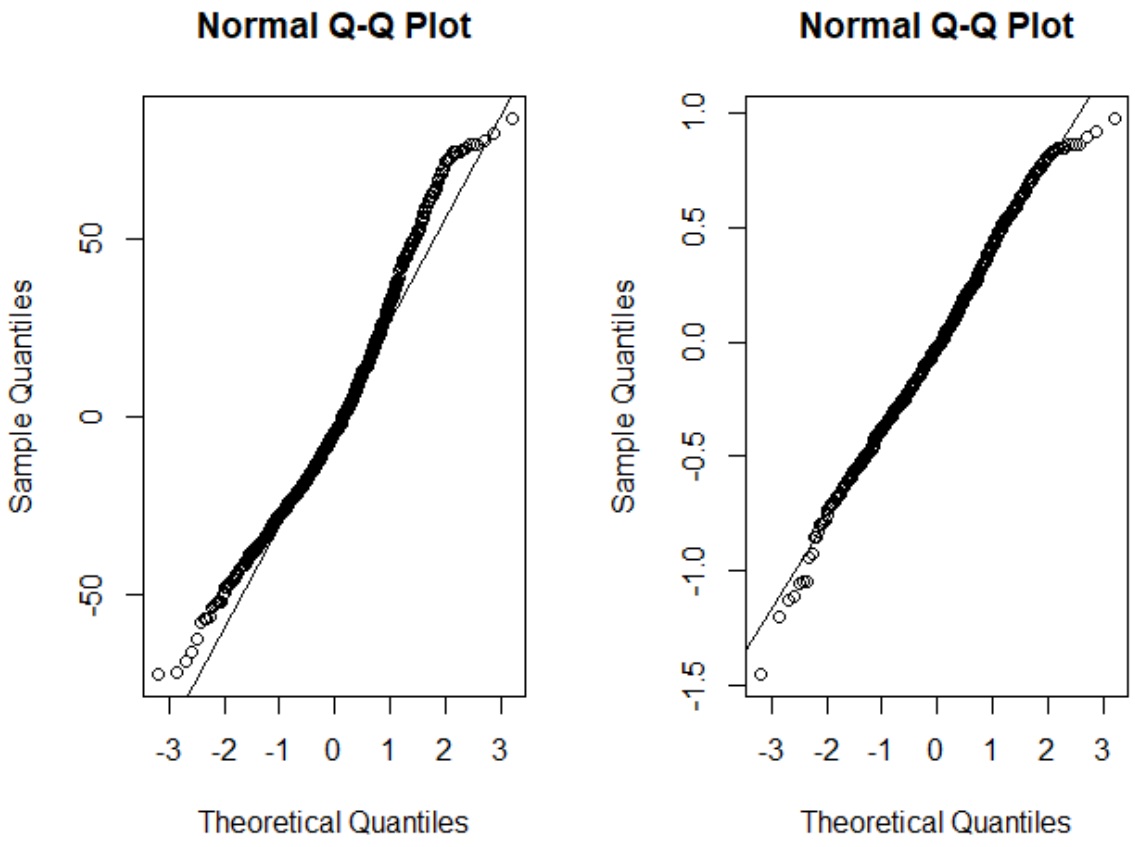

Figure 4.2 – Q-Q plot before and after transformation

```
> reg1 <- lm(Pima_NA$glucose~Pima_NA$pressure)
> transf <- boxcox(Pima_NA$glucose~Pima_NA$pressure)
> lambda <- transf$x[which.max(transf$y)]
> reg2 <- lm(((Pima_NA$glucose^lambda-1)/lambda)~Pima_NA$pressure)
>
> # Q-Q plots.
> par(mfrow = c(1,2))
> qqnorm(reg1$residuals)
> qqline(reg1$residuals)
> qqnorm(reg2$residuals)
> qqline(reg2$residuals)
```

We can see from figure 4.2 that the data now closely follows the line defining the normal distribution. In the first graph, the data points are more scattered, leaning away from the normal distribution. But that isn't the case in the second graph, after applying the Box-Cox transformation. In order for data to follow a normal distribution, it should be close to the normal line on the Q-Q plot. So in this case, we can say the transformation worked in getting the data closer to a normal distribution.

Lastly, we can perform the Shapiro-Wilk test again to check the new p-value.

```
> shapiro.test((Pima_NA$glucose^lambda-1)/lambda)
     Shapiro-Wilk normality test
data:  (Pima_NA$glucose^lambda - 1)/lambda
W = 0.99138, p-value = 0.0001978
>
```

Clearly, the p-value has increased significantly from the previous one, and is now in the range $0.001 < p < 0.01$. However, it's still not large enough to prove the normality hypothesis. This concludes that the dataset cannot be generated from a normal distribution.

5) Summary of Investigation:

This report looked into the dataset PimaIndiansDiabetes and worked on pre-processing it for further analysis.

The data had many missing values as well as outliers in most of the continuous columns, but not all of them. These values were set to "NA" before being imputed as the mean.

Once the data was cleaner, it was normalised using the Z-score, and the null normality hypothesis was defined. Using the Shapiro-Wilk test determined that the data needed transforming in order to follow a normal distribution, as the p-value was too small, almost close to 0. After a Box-Cox transformation, the p-value increased significantly, and Q-Q plots indicated that the dataset is following a normal distribution a lot closer than it was previously. However, the new p-value is still relatively low, so it is a high possibility that the data indeed cannot be generated from a normally distributed sample.

6) Conclusion:

Lack of awareness about the dangers of diabetes, combined with inadequate access to health services, could be fatal. It is a universal problem with overwhelming human, social, and economic impact, affecting around 300 million people worldwide. Many nations are facing the rapid growth of diabetes

among their citizens, and the WHO already warned that diabetes will be the 7th leading cause of death in the world by 2030.

By applying computational analytics on clinical data, the massive amount of information generated in the healthcare systems can be used to create medical intelligence which can drive the sector forward, reduce medical costs, and help create more patient-centred healthcare systems.

Bibliography:

Baier, L.J. and Hanson, R.L. (2004). Genetic Studies of the Etiology of Type 2 Diabetes in Pima Indians: Hunting for Pieces to a Complicated Puzzle. *Diabetes*, [online] 53(5), pp.1181–1186. Available at: https://diabetes.diabetesjournals.org/content/53/5/1181 [Accessed 25 Nov. 2020].

Box, G.E.P. and Cox, D.R. (1964). An Analysis of Transformations. *Journal of the Royal Statistical Society: Series B (Methodological)*, 26(2), pp.211–243.

Leisch, F. (n.d.). *R: Pima Indians Diabetes Database*. [online] www.cit.ctu.edu.vn. Available at: http://www.cit.ctu.edu.vn/~dtnghi/detai/PimaIndiansDiabetes.html [Accessed 23 Nov. 2020].

Venkat Narayan, K.M. (1997). Diabetes Mellitus in Native Americans: The Problem and Its Implications. *Population Research and Policy Review*, 16(1/2), pp.169–192.

WHO (2016). *World Health Day 2016: Beat diabetes*. [online] WHO. Available at: https://www.who.int/campaigns/world-health-day/2016/event/en/#:~:text=In%202012%2C%20the%20disease%20was,cause%20of%20death%20by%202030. [Accessed 27 Nov. 2020].