Foundations of Data Science Project - Diabetes Analysis

Context

This dataset is originally from the National Institute of Diabetes and Digestive and Kidney Diseases. The objective is to predict based on diagnostic measurements whether a patient has diabetes.

Diabetes is one of the most frequent diseases worldwide and the number of diabetic patients are growing over the years. The main cause of diabetes remains unknown, yet scientists believe that both genetic factors and environmental lifestyle play a major role in diabetes.

A few years ago research was done on a tribe in America which is called the Pima tribe (also known as the Pima Indians). In this tribe, it was found that the ladies are prone to diabetes very early. Several constraints were placed on the selection of these instances from a larger database. In particular, all patients were females at least 21 years old of Pima Indian heritage.

Objective

Analyzing different aspects of Diabetes in the Pima Indians tribe by doing Exploratory Data Analysis.

Dataset information:

- Pregnancies: Number of times pregnant
- Glucose: Plasma glucose concentration over 2 hours in an oral glucose tolerance test
- BloodPressure: Diastolic blood pressure (mm Hg)
- SkinThickness: Triceps skin fold thickness (mm)
- Insulin: 2-Hour serum insulin (mu U/ml)
- BMI: Body mass index (weight in kg/(height in m)^2)
- DiabetesPedigreeFunction: A function which scores likelihood of diabetes based on family history.
- Age: Age in years
- Outcome : Class variable (0: person is not diabetic or 1: person is diabetic)

Importing libraries

In [39]:

```
import numpy as np
import pandas as pd

import seaborn as sns
import matplotlib.pyplot as plt
```

Importing data and naming table "pima"

In [8]:

```
pima = pd.read_csv(r'C:\Users\Tengetile Nxumalo\Downloads\diabetes.csv')
```

In [10]:

```
#Viewing top 10 entries of table
pima.head(10)
```

Out[10]:

	Pregnancies	Glucose	BloodPressure	SkinThickness	Insulin	BMI	DiabetesPedigreeFunc
0	6	148	72	35	0	33.6	0.
1	1	85	66	29	0	26.6	0.
2	8	183	64	0	0	23.3	0.
3	1	89	66	23	94	28.1	0.
4	0	137	40	35	168	43.1	2.
5	5	116	74	0	0	25.6	0.
6	3	78	50	32	88	31.0	0.
7	10	115	0	0	0	35.3	0.
8	2	197	70	45	543	30.5	0.
9	8	125	96	0	0	0.0	0.

In [11]:

```
#Viewing bottom 10 entries of table
pima.tail(10)
```

Out[11]:

	Pregnancies	Glucose	BloodPressure	SkinThickness	Insulin	BMI	DiabetesPedigreeFu
758	1	106	76	0	0	37.5	
759	6	190	92	0	0	35.5	
760	2	88	58	26	16	28.4	
761	9	170	74	31	0	44.0	
762	9	89	62	0	0	22.5	
763	10	101	76	48	180	32.9	
764	2	122	70	27	0	36.8	
765	5	121	72	23	112	26.2	
766	1	126	60	0	0	30.1	
767	1	93	70	31	0	30.4	

Checking at the dimensions of the datase

In [65]:

```
pima.shape
```

Out[65]:

```
(768, 9)
```

768 rows and 9 columns

Data types of all the variables in the data set

In [15]:

```
pima.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 768 entries, 0 to 767
Data columns (total 9 columns):
 #   Column                    Non-Null Count  Dtype  
---  ------                    --------------  -----  
 0   Pregnancies               768 non-null    int64  
 1   Glucose                   768 non-null    int64  
 2   BloodPressure             768 non-null    int64  
 3   SkinThickness             768 non-null    int64  
 4   Insulin                   768 non-null    int64  
 5   BMI                       768 non-null    float64
 6   DiabetesPedigreeFunction  768 non-null    float64
 7   Age                       768 non-null    int64  
 8   Outcome                   768 non-null    int64  
dtypes: float64(2), int64(7)
memory usage: 54.1 KB
```

Combination of integer and float numerical data. Integers have no decinmal points while float-point have decimals.

Checking if we have missing values

In [16]:

```
pima.isnull().values.any()
```

Out[16]:

```
False
```

no missing value/ entry

Summary statistics of all variables except the last column (Outcome)

In [17]:

```
pima.iloc[:,0:8].describe()
```

Out[17]:

	Pregnancies	Glucose	BloodPressure	SkinThickness	Insulin	BMI	Diab
count	768.000000	768.000000	768.000000	768.000000	768.000000	768.000000	
mean	3.845052	120.894531	69.105469	20.536458	79.799479	31.992578	
std	3.369578	31.972618	19.355807	15.952218	115.244002	7.884160	
min	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	
25%	1.000000	99.000000	62.000000	0.000000	0.000000	27.300000	
50%	3.000000	117.000000	72.000000	23.000000	30.500000	32.000000	
75%	6.000000	140.250000	80.000000	32.000000	127.250000	36.600000	
max	17.000000	199.000000	122.000000	99.000000	846.000000	67.100000	

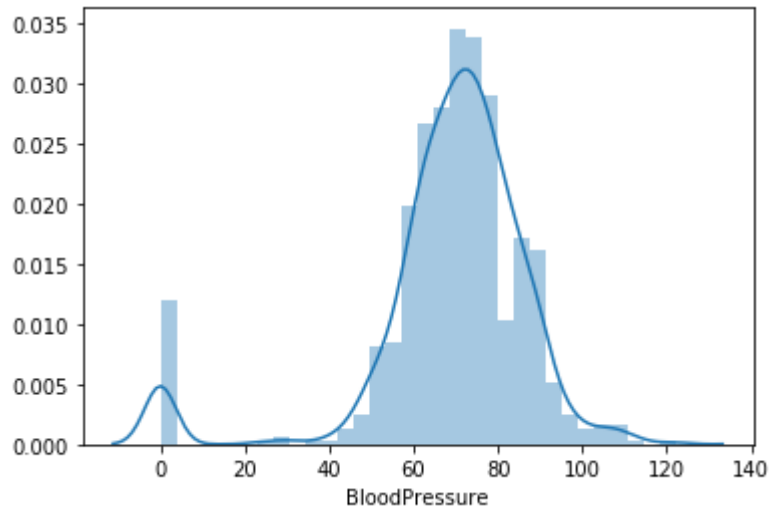
Summary statistcs is a table which gives a summary of the descriptive statistics for each of the columns.For example, the Glucose column, the count is the number of entries which is 768. Mean is the average, which is 120.89, std is the standard deviation which measures the dispersion of the dataset from the mean, this is 31.97. Min is the minimum value of glucose recorded and is 0. Max, is the maximum Glucose value recorded, which is 199.0. 25% is the 25th percentile which is 99.0, while 75% is the 75th percentile or upper percentile which is 140.25. 50% is the 50th percentile which is the same as the median, this is 117.0.

Data Exploration

Distribution of blood pressure

In [23]:

```
sns.distplot(pima['BloodPressure'], kde = True) #kde = kernel density estimate
plt.show()
```



The plot is showing a normal distribution with the outlier of blood pressure at abou zero. The median at about 70 also coincides with the mode. A majority of the sample data falls under the population curve.

Information that can be extractes from dataset

Calculating the three measures of central tendency.

In [31]:

```
m1 = pima['BMI'].mean()
m2 = pima['BMI'].median()
m3 = pima['BMI'].mode()[0]
print(m1)
print(m2)
print(m3)
```

```
31.992578124999977
32.0
32.0
```

mean = 32.0, median = 32.0, mode = 32.0.

In [35]:

```
#Number of  women with 'Glucose' level above the mean level of 'Glucose'

pima[pima['Glucose']>pima['Glucose'].mean()].shape[0]
```

Out[35]:

```
349
```

In [36]:

```
#Showing entries of women that have their 'BloodPressure' equal to the median of 'Blood
Pressure' and their 'BMI' less than the median of 'BMI'
pima[(pima['BloodPressure']==pima['BloodPressure'].median()) & (pima['BMI']<pima['BMI']
.median())]
```

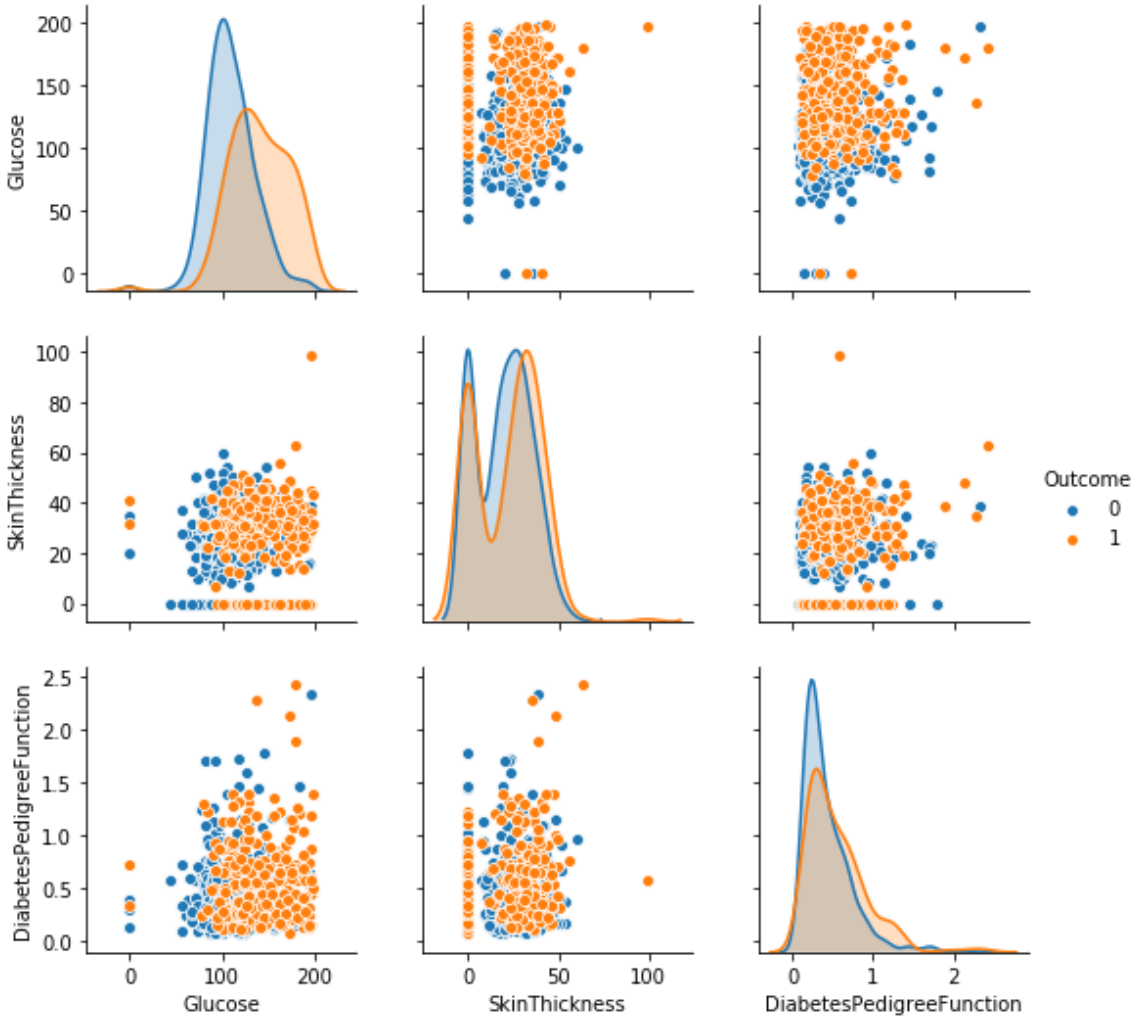
Out[36]:

	Pregnancies	Glucose	BloodPressure	SkinThickness	Insulin	BMI	DiabetesPedigreeFu
14	5	166	72	19	175	25.8	
93	4	134	72	0	0	23.8	
103	1	81	72	18	40	26.6	
205	5	111	72	28	0	23.9	
299	8	112	72	0	0	23.6	
325	1	157	72	21	168	25.6	
330	8	118	72	19	0	23.1	
366	6	124	72	0	0	27.6	
380	1	107	72	30	82	30.8	
393	4	116	72	12	87	22.1	
406	4	115	72	0	0	28.9	
446	1	100	72	12	70	25.3	
460	9	120	72	22	56	20.8	
488	4	99	72	17	0	25.6	
497	2	81	72	15	76	30.1	
510	12	84	72	31	0	29.7	
568	4	154	72	29	126	31.3	
615	3	106	72	0	0	25.8	
635	13	104	72	0	0	31.2	
644	3	103	72	30	152	27.6	
717	10	94	72	18	0	23.1	
765	5	121	72	23	112	26.2	

Creating pairplot for variables 'Glucose', 'SkinThickness' and 'DiabetesPedigreeFunction'. Paiplots help show pairwise relationships in a dataset. A third variable can be added on a pairplot in a form of a hue. In this case we will add the variable 'Outcome' to each of the plots.

In [37]:

```
sns.pairplot(data=pima,vars=['Glucose', 'SkinThickness', 'DiabetesPedigreeFunction'], h
ue='Outcome')
plt.show()
```

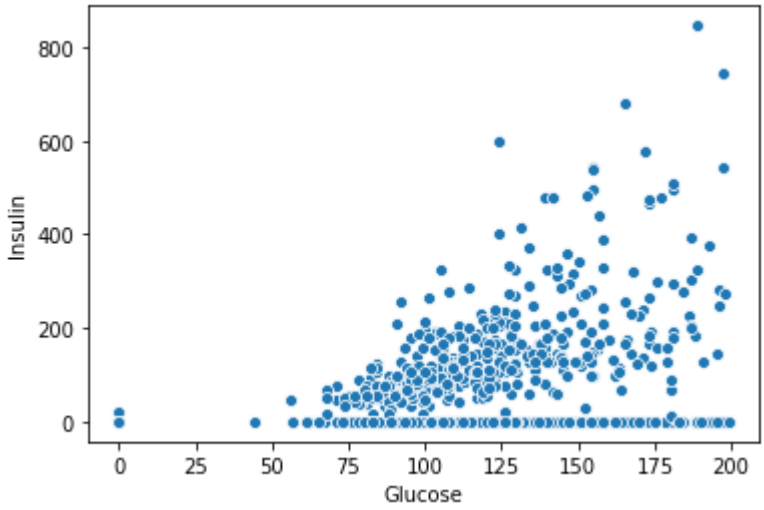


Glucose levels tend to be higher for patients with diabetes. The DiabetesPedigreeFunction does not seem to be correlated to whether the patient will have diabetes or not so no particular conclusion can be made from this plot.Also, no obvious correlations between the the three variables

Scatterplot between 'Glucose' and 'Insulin'

In [40]:

```
sns.scatterplot(x='Glucose',y='Insulin',data=pima)
plt.show()
```

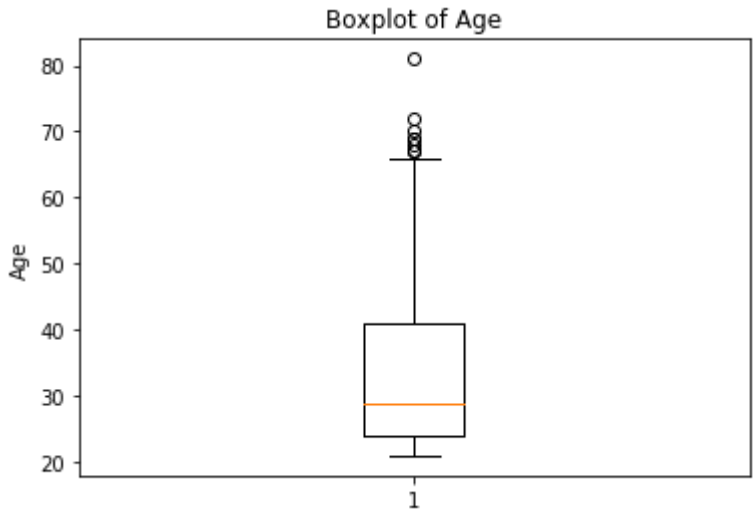


There is some positive correlation between the glucose and insulin

Boxplot for 'Age'. Advantage of boxplots is that they show outliers , which are data points that differ significantly from the rest of the dataset. Boxplots demonstrate the spread and skewness groups of numerical data through quartiles.

In [41]:

```
plt.boxplot(pima['Age'])

plt.title('Boxplot of Age')
plt.ylabel('Age')
plt.show()
```

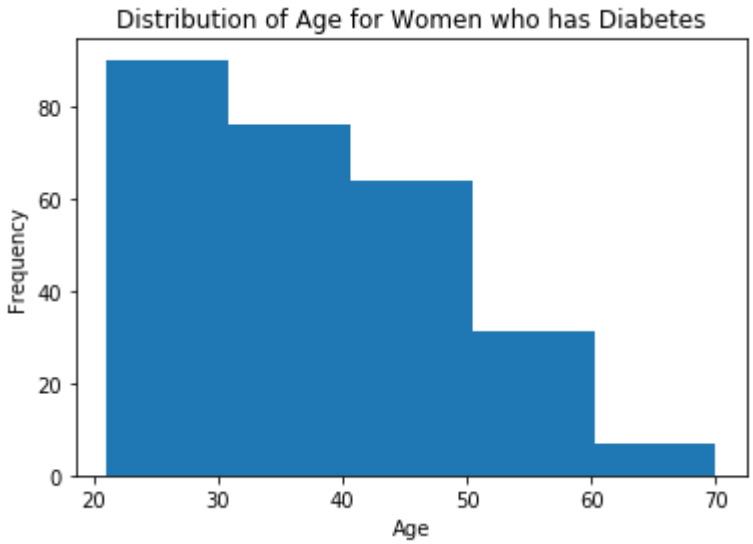


We have a few outliers, outside the maximum whisker. Females with age > 65

Plotting histograms for variable Age to understand the number of women in different Age groups given that they have diabetes or not.

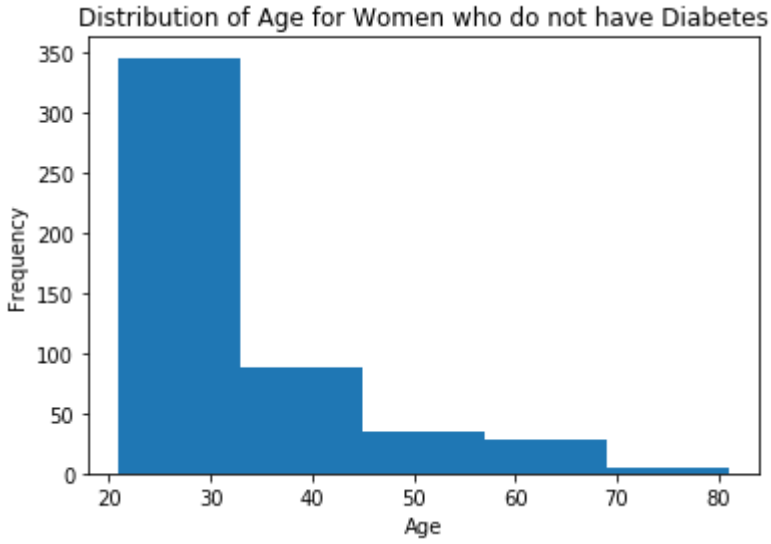
In [45]:

```
plt.hist(pima[pima['Outcome']==1]['Age'], bins = 5)
plt.title('Distribution of Age for Women who has Diabetes')
plt.xlabel('Age')
plt.ylabel('Frequency')
plt.show()
```



In [47]:

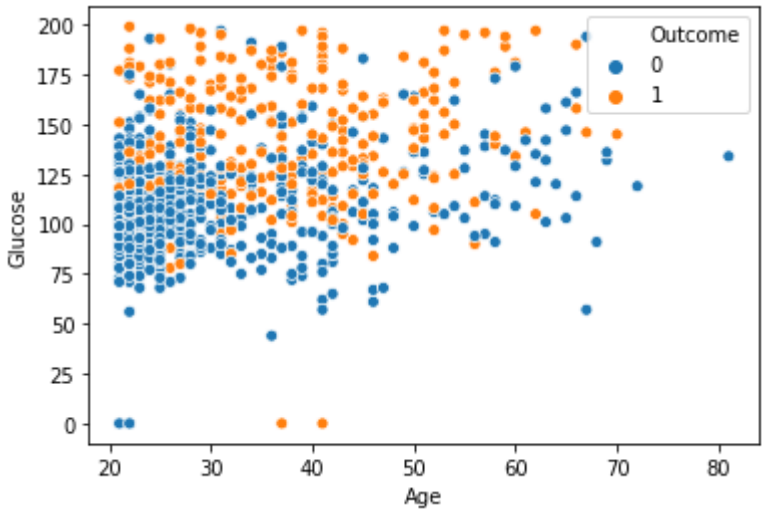
```
plt.hist(pima[pima['Outcome']==0]['Age'], bins = 5)
plt.title('Distribution of Age for Women who do not have Diabetes')
plt.xlabel('Age')
plt.ylabel('Frequency')
plt.show()
```



Since the number of women tested for the different age groups is not the same, it would not be useful to compare the frequencies for the different plots. For example, even though the age group between 20 and 30 years is showing the highest frequency for the womenn with diabetes, this is also the case for the women without diabetes. The lower age group between 20 and 30 has the lowest ratio of women with diabetes and this ratio increses with age. Ratio of women with diabetes is lowest for the 20-30 years age group. This can also be seen on the scatterplot below.

In [49]:

```
sns.scatterplot(x='Age',y='Glucose',data=pima, hue = 'Outcome')
plt.show()
```



Finding and visualizing the correlation matrix for non categorical variables. The matrix depicts the correlation between all the possible pairs of values

In [52]:

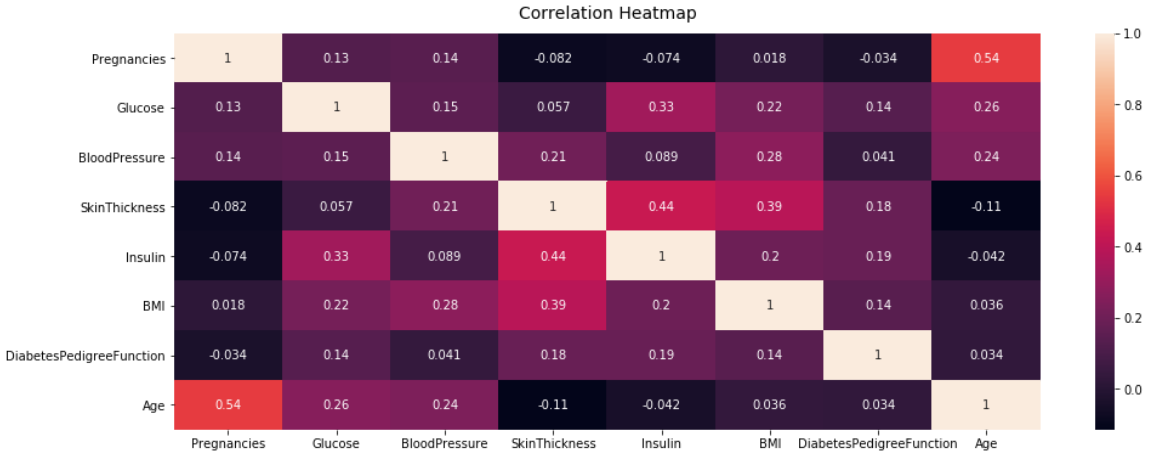
```
corr_matr = pima.iloc[:,0:8].corr()

corr_matr
```

Out[52]:

	Pregnancies	Glucose	BloodPressure	SkinThickness	Insulin	
Pregnancies	1.000000	0.129459	0.141282	-0.081672	-0.073535	
Glucose	0.129459	1.000000	0.152590	0.057328	0.331357	
BloodPressure	0.141282	0.152590	1.000000	0.207371	0.088933	
SkinThickness	-0.081672	0.057328	0.207371	1.000000	0.436783	
Insulin	-0.073535	0.331357	0.088933	0.436783	1.000000	
BMI	0.017683	0.221071	0.281805	0.392573	0.197859	
DiabetesPedigreeFunction	-0.033523	0.137337	0.041265	0.183928	0.185071	
Age	0.544341	0.263514	0.239528	-0.113970	-0.042163	

In [61]:

```
plt.figure(figsize=(16, 6))
heatmap = sns.heatmap(corr_matr, annot = True)
bottom, top = heatmap.get_ylim()
heatmap.set_ylim(bottom + 0.5, top - 0.5)
heatmap.set_title('Correlation Heatmap', fontdict={'fontsize':14}, pad=12)
plt.show()
```



Pregnancies and age are showing the highest positive correlation. SkinThickness is also showing a positive correlation with insulin and BMI, which is a weak association from the value of correlation coefficient.