# EDA and Regression Coursework

Student ID: 10724837

## 1 The Data

This dataset records 8 diagnostic measures, for 750 women, which may be used to predict whether or not they will develop diabetes at some point.

The data has 9 columns and 750 entries. All columns are of integer data type, except BMI and DiabetesPedigree, which are of float. Outcome could be converted to a boolean data type, but we can work with 0s and 1s just the same.

There are no null values or negative numbers throughout the dataset. The maximum values in each column are also reasonable.

However, there are zero values in columns where there should not be; for example, there are zero values in the SkinThickness and BMI columns, which should not be possible. There are also a lot of them; in the SkinThickness column, there are 221 zeroes out of the 750 total observations, which is a considerable amount.

Apart from this issue, the data is clean.

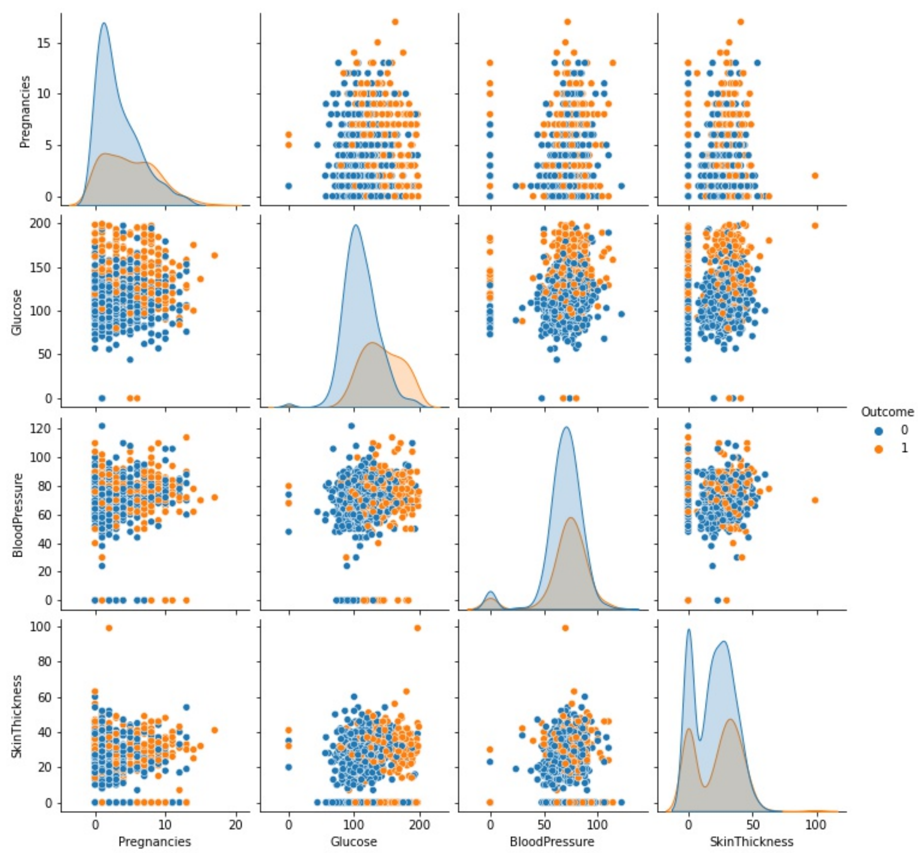
## 2 Exploratory Data Analysis

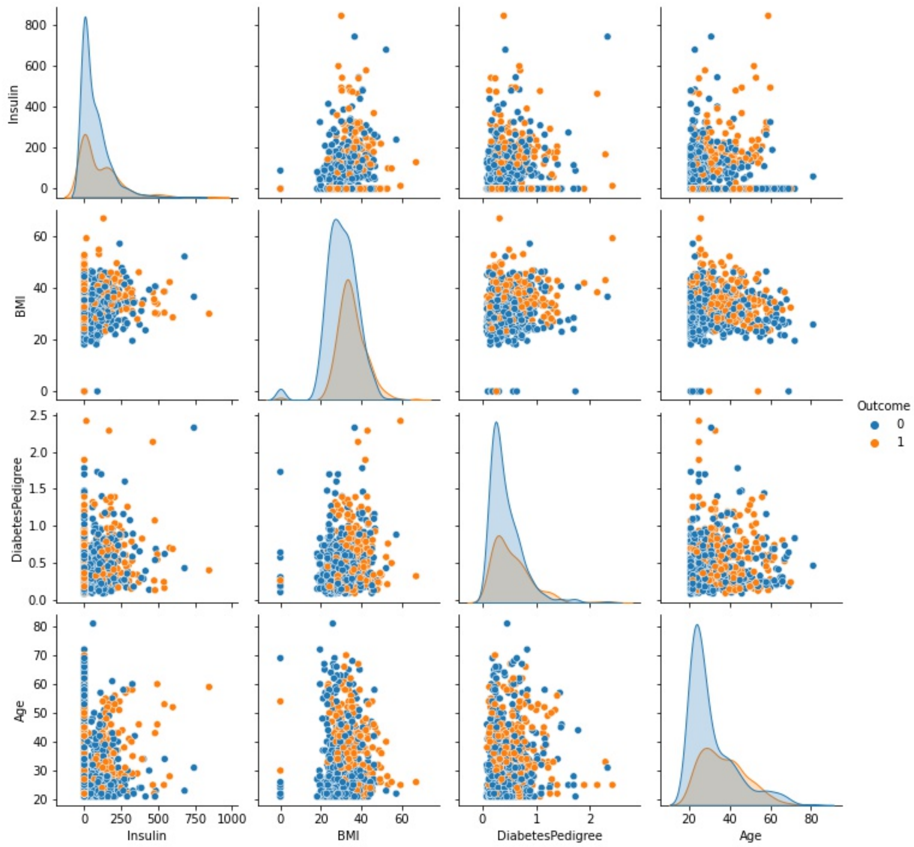
I complete the EDA within Python.

Checking the correlations between the variables, there are no outstandingly high values, so no strong conclusions can be made here. The strongest correlations are between Age and Pregnancies (0.547124), and Glucose and Outcome (0.460310). The first implies that it is unlikely that we will include both Age and Pregnancies together in our final model. The second implies that Glucose is likely to be included in our final model, as it directly correlates to the Outcome more strongly than the other variables.

| | Pregnancies | Glucose | BloodPressure | SkinThickness | Insulin | BMI | DiabetesPedigree | Age | Outcome |
|---|---|---|---|---|---|---|---|---|---|
| **Pregnancies** | 1.000000 | 0.129594 | 0.142453 | -0.087047 | -0.070822 | 0.021739 | -0.031085 | 0.547124 | 0.229235 |
| **Glucose** | 0.129594 | 1.000000 | 0.145972 | 0.056647 | 0.333005 | 0.214316 | 0.140364 | 0.259797 | 0.460310 |
| **BloodPressure** | 0.142453 | 0.145972 | 1.000000 | 0.205494 | 0.086750 | 0.278569 | 0.042922 | 0.237693 | 0.060860 |
| **SkinThickness** | -0.087047 | 0.056647 | 0.205494 | 1.000000 | 0.436093 | 0.394615 | 0.189191 | -0.115862 | 0.082205 |
| **Insulin** | -0.070822 | 0.333005 | 0.086750 | 0.436093 | 1.000000 | 0.195726 | 0.191289 | -0.040152 | 0.130928 |
| **BMI** | 0.021739 | 0.214316 | 0.278569 | 0.394615 | 0.195726 | 1.000000 | 0.143798 | 0.032972 | 0.289832 |
| **DiabetesPedigree** | -0.031085 | 0.140364 | 0.042922 | 0.189191 | 0.191289 | 0.143798 | 1.000000 | 0.041807 | 0.170688 |
| **Age** | 0.547124 | 0.259797 | 0.237693 | -0.115862 | -0.040152 | 0.032972 | 0.041807 | 1.000000 | 0.232892 |
| **Outcome** | 0.229235 | 0.460310 | 0.060860 | 0.082205 | 0.130928 | 0.289832 | 0.170688 | 0.232892 | 1.000000 |

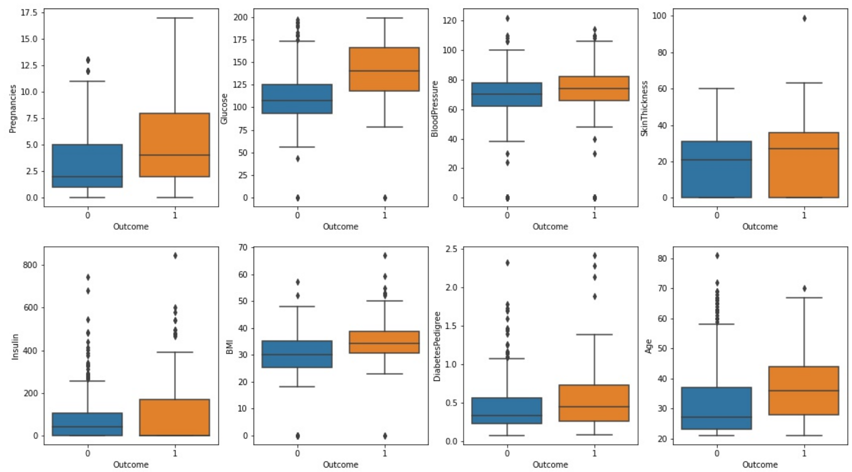
We use pairplots to check the distributions of each variable, as well as some of the relationships between them. They all tend to follow the normal distribution, with many of them being positively skewed. This may negatively affect the accuracy of our predictions later on. We could have used Box-Cox transformations to deal with this.






SkinThickness has a bimodal distribution. Although this would normally need to be considered, this variable is not included in our final model anyway, so this can be ignored.

We use boxplots to see how each variable differs between women who eventually end up with diabetes, and those who do not. We find that BloodPressure and SkinThickness have less effect on the Outcome, compared to the other variables (this could also be seen in the correlations output). This makes them candidates for removal in our regression model.




## 3 SevenOrMorePregnancies

I add the SevenOrMorePregnancies column in Python, save the edited .csv file, then switch to R for the rest of the analysis.

Since Outcome is a binary categorical variable, the appropriate choice is a logistic regression model, with formula $Outcome \sim SevenOrMorePregnancies$. We get the intercept and gradient values as -0.91988 and 1.18980 respectively. Using these values, we can state the regression equation, then work out the probabilities requested in the brief. Here is the working:

Let

$p$ = probability that a woman eventually tests positive for diabetes

$x$ = SevenOrMorePregnancies $= \begin{cases} 0 & \text{if False} \\ 1 & \text{if True} \end{cases}$

Using the estimates $\beta_0 = -0.91988$ and $\beta_1 = 1.18980$ given by our logistic regression in R, the estimated model is

$$\log\left(\frac{p}{1-p}\right) = -0.91988 + 1.18980x.$$

$$\Rightarrow \frac{p}{1-p} = \exp(-0.91988 + 1.18980x)$$

$$\Rightarrow p = (1-p)\exp(-0.91988 + 1.18980x)$$

$$\Rightarrow p\left(1 + \exp(-0.91988 + 1.18980x)\right) = \exp(-0.91988 + 1.18980x)$$

$$\Rightarrow p = \frac{\exp(-0.91988 + 1.18980x)}{1 + \exp(-0.91988 + 1.18980x)}$$

The probability that you get diabetes, given that you have six or fewer pregnancies, is given by $p$ when $x = 0$.

$$p = 0.2849823458$$

For seven or more pregnancies, $x = 1$ and

$$p = 0.567073265$$

We end up with probabilities 0.2849823458 and 0.567073265.

# 4 Final Regression Model

When developing a regression model, we must attempt to pick the predictors which best describe the dependent variable, whilst keeping the model as simple as possible. A good, yet not conclusive, indicator of a model's accuracy is the adjusted $R^2$ statistic. $R^2$ describes the proportion of variance in the dependent variable which is described by the independent variables. The drawback of $R^2$ is that including extra predictors will only ever improve the $R^2$. The adjusted $R^2$ statistic, on the other hand, penalises the inclusion of extra predictors.

We also do not want too many predictors in the model, as this can lead to overfitting and introduces unnecessary complexity for a simple prediction! Generally, we try to strike a balance between the two.

To do this, I opted for a method known as "backward stepwise regression", where we start with a model including all predictors, then eliminate the least statistically significant predictor, one-by-one.

With all predictors, we get an adjusted $R^2$ of 0.2914. The least significant predictor was SkinThickness, with a p-value of 0.65636.

Removing SkinThickness and trying again, we get an adjusted $R^2$ of 0.2922, an improvement. Thus, we can confidently remove SkinThickness from the model. The least significant predictor this time was Insulin.

Removing Insulin and running it again, we get an adjusted $R^2$ of 0.2918, a negligible decrease. We remove Insulin in order to simplify the model, without much loss in accuracy. The next least significant predictor was Age.

Removing Age, we get an adjusted $R^2$ of 0.2905. This is a small, but not quite negligible, decrease. Earlier, we saw that Age and Pregnancies had a moderate correlation (0.547124 specifically). Since it is unlikely that both of these will be included in the model, try removing Pregnancy instead of Age and see what the effect is.

On doing so, we get an adjusted $R^2$ of 0.2748, a significant decrease. This implies that we should remove Age and keep Pregnancies.

At this point, there are no obvious candidates for removal, yet the model is still quite complex. We saw earlier in the Python boxplots and the correlation table that BloodPressure did not seem a great predictor for Outcome. So we try removing this.

With BloodPressure removed, we get an adjusted $R^2$ of 0.2848, which is a decrease, but a necessary one for the simplification of the model. From here, any further removal of predictors leads to drastic drops in the adjusted $R^2$, so the remaining 4 predictors make up our final model.

Reading off the values of the output in R, we obtain the following regression equation:

$$y = -0.8678511 + 0.0252114x_1 + 0.0056975x_2 + 0.0114245x_3 + 0.1363881x_4,$$

where $y$ = Outcome, $x_1$ = Pregnancies, $x_2$ = Glucose, $x_3$ = BMI and $x_4$ = DiabetesPedigree.

Finally, we load the ToPredict.csv file, then use the predict function with our model to predict the probabilities that those women will eventually end up with diabetes. We get the following probabilities; 0.5255120, 0.3279125, 0.1505665, 0.6883586 and 0.6418316. (Although when using the above regression equation and checking by hand, we end up with values which are very slightly different, likely due to rounding in computer calculations).

# 5 Appendix

## 5.1 Python Code

I apologise for the poor formatting here. On the following pages, I include my Jupyter notebook and R code.

In [1]:
```
########
# Student ID: 10724837
# In this notebook, I carry out some basic observation of the data, as well as the E
# before moving over to R to develop the regression models.
########
```

In [2]:
```
# import packages we will be using
import pandas as pd
import numpy as np
import seaborn as sns
import matplotlib.pyplot as plt
```

In [3]:
```
# load the data and observe basic info
diabetes = pd.read_csv("PimaDiabetes.csv")
diabetes.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 750 entries, 0 to 749
Data columns (total 9 columns):
 #   Column            Non-Null Count  Dtype
---  ------            --------------  -----
 0   Pregnancies       750 non-null    int64
 1   Glucose           750 non-null    int64
 2   BloodPressure     750 non-null    int64
 3   SkinThickness     750 non-null    int64
 4   Insulin           750 non-null    int64
 5   BMI               750 non-null    float64
 6   DiabetesPedigree  750 non-null    float64
 7   Age               750 non-null    int64
 8   Outcome           750 non-null    int64
dtypes: float64(2), int64(7)
memory usage: 52.9 KB
```

In [4]:
```
# check for null values
diabetes.isna().sum(axis = 0)
```

Out[4]:
```
Pregnancies         0
Glucose             0
BloodPressure       0
SkinThickness       0
Insulin             0
BMI                 0
DiabetesPedigree    0
Age                 0
Outcome             0
dtype: int64
```

In [5]:
```
# there are no null values!
```

In [6]:
```
# calculate some key summary statistics
diabetes.describe()
```

Out[6]:

| | Pregnancies | Glucose | BloodPressure | SkinThickness | Insulin | BMI | DiabetesPedig |
|---|---|---|---|---|---|---|---|
| **count** | 750.000000 | 750.000000 | 750.000000 | 750.000000 | 750.000000 | 750.000000 | 750.000( |
| **mean** | 3.844000 | 120.737333 | 68.982667 | 20.489333 | 80.378667 | 31.959067 | 0.473! |

| | Pregnancies | Glucose | BloodPressure | SkinThickness | Insulin | BMI | DiabetesPedig |
|---|---|---|---|---|---|---|---|
| **std** | 3.370085 | 32.019671 | 19.508814 | 15.918828 | 115.019198 | 7.927399 | 0.332 |
| **min** | 0.000000 | 0.000000 | 0.000000 | 0.000000 | 0.000000 | 0.000000 | 0.078 |
| **25%** | 1.000000 | 99.000000 | 62.000000 | 0.000000 | 0.000000 | 27.300000 | 0.244 |
| **50%** | 3.000000 | 117.000000 | 72.000000 | 23.000000 | 36.500000 | 32.000000 | 0.377 |
| **75%** | 6.000000 | 140.750000 | 80.000000 | 32.000000 | 129.750000 | 36.575000 | 0.628 |
| **max** | 17.000000 | 199.000000 | 122.000000 | 99.000000 | 846.000000 | 67.100000 | 2.420 |

In [7]:

```
# note that none of the minimums are less than 0, and none of the maximums are unrea
```

In [8]:

```
# check the number of zero values in each column
diabetes.isin([0]).sum(axis = 0)
```

Out[8]:

```
Pregnancies          109
Glucose                5
BloodPressure         35
SkinThickness        221
Insulin              362
BMI                   11
DiabetesPedigree       0
Age                    0
Outcome              490
dtype: int64
```

In [9]:

```
# there are lots of zeroes in columns where it doesn't make sense
# in particular, you cannot have a SkinThickness or BMI of 0
# this is a problem with the data
```

In [10]:

```
# calculate the covariance matrix
diabetes.cov()
```

Out[10]:

| | Pregnancies | Glucose | BloodPressure | SkinThickness | Insulin | BMI |
|---|---|---|---|---|---|---|
| **Pregnancies** | 11.357474 | 13.984336 | 9.365784 | -4.669891 | -27.452198 | 0.580789 |
| **Glucose** | 13.984336 | 1025.259352 | 91.183692 | 28.873696 | 1226.417353 | 54.400449 |
| **BloodPressure** | 9.365784 | 91.183692 | 380.593825 | 63.817572 | 194.658108 | 43.081800 |
| **SkinThickness** | -4.669891 | 28.873696 | 63.817572 | 253.409098 | 798.472669 | 49.798428 |
| **Insulin** | -27.452198 | 1226.417353 | 194.658108 | 798.472669 | 13229.415833 | 178.463985 |
| **BMI** | 0.580789 | 54.400449 | 43.081800 | 49.798428 | 178.463985 | 62.843649 |
| **DiabetesPedigree** | -0.034792 | 1.492680 | 0.278102 | 1.000246 | 7.307274 | 0.378596 |
| **Age** | 21.589453 | 97.401647 | 54.295283 | -21.595683 | -54.073876 | 3.060503 |
| **Outcome** | 0.367904 | 7.019083 | 0.565429 | 0.623195 | 7.171624 | 1.094182 |

In [11]:

```
# calculate the correlation matrix
diabetes.corr()
```

Out[11]:

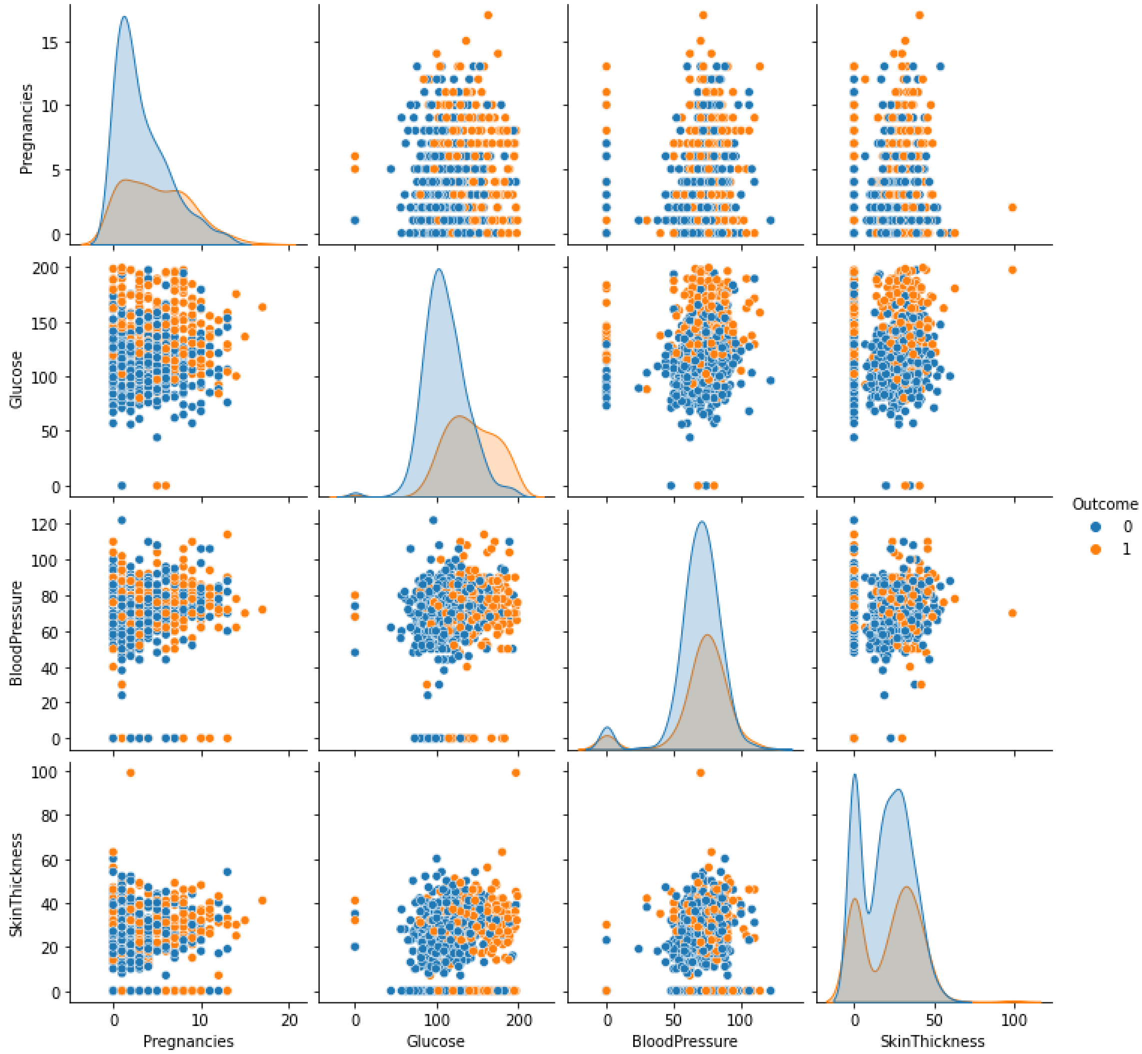
| | Pregnancies | Glucose | BloodPressure | SkinThickness | Insulin | BMI | Diabetes |
|---|---|---|---|---|---|---|---|
| **Pregnancies** | 1.000000 | 0.129594 | 0.142453 | -0.087047 | -0.070822 | 0.021739 | - |
| **Glucose** | 0.129594 | 1.000000 | 0.145972 | 0.056647 | 0.333005 | 0.214316 | |
| **BloodPressure** | 0.142453 | 0.145972 | 1.000000 | 0.205494 | 0.086750 | 0.278569 | |
| **SkinThickness** | -0.087047 | 0.056647 | 0.205494 | 1.000000 | 0.436093 | 0.394615 | |
| **Insulin** | -0.070822 | 0.333005 | 0.086750 | 0.436093 | 1.000000 | 0.195726 | |
| **BMI** | 0.021739 | 0.214316 | 0.278569 | 0.394615 | 0.195726 | 1.000000 | |
| **DiabetesPedigree** | -0.031085 | 0.140364 | 0.042922 | 0.189191 | 0.191289 | 0.143798 | |
| **Age** | 0.547124 | 0.259797 | 0.237693 | -0.115862 | -0.040152 | 0.032972 | |
| **Outcome** | 0.229235 | 0.460310 | 0.060860 | 0.082205 | 0.130928 | 0.289832 | |

In [12]:

```
# the strongest correlations are between age and pregnancies
# and between glucose and outcome
# all columns are somewhat uncorrelated, there are no clear candidates for columns w
```

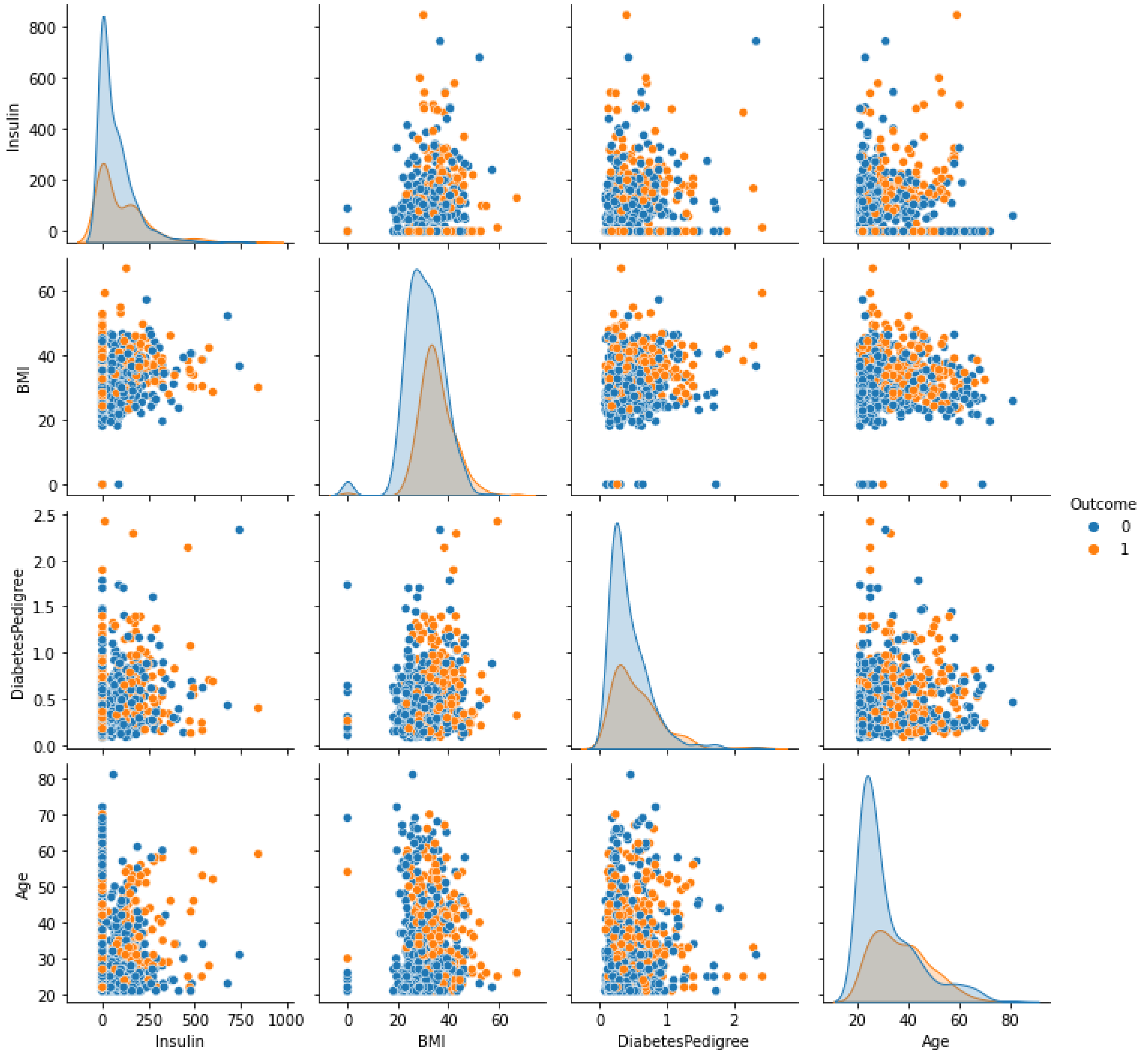
In [13]:

```
# we want to check the distribution of each columns
# one way to do this is by using pairplots on each column
# do 4 columns at a time for the sake of presentability
sns.pairplot(data = diabetes[['Pregnancies','Glucose','BloodPressure','SkinThickness

# save the figure
plt.savefig("pythonImage1.jpg")
```




In [14]:
```
# note that skin thickness is bimodal
```

In [15]:
```
# pairplots for the remaining 4
sns.pairplot(data = diabetes[['Insulin','BMI','DiabetesPedigree','Age','Outcome']],

# save the figure
plt.savefig("pythonImage2.jpg")
```



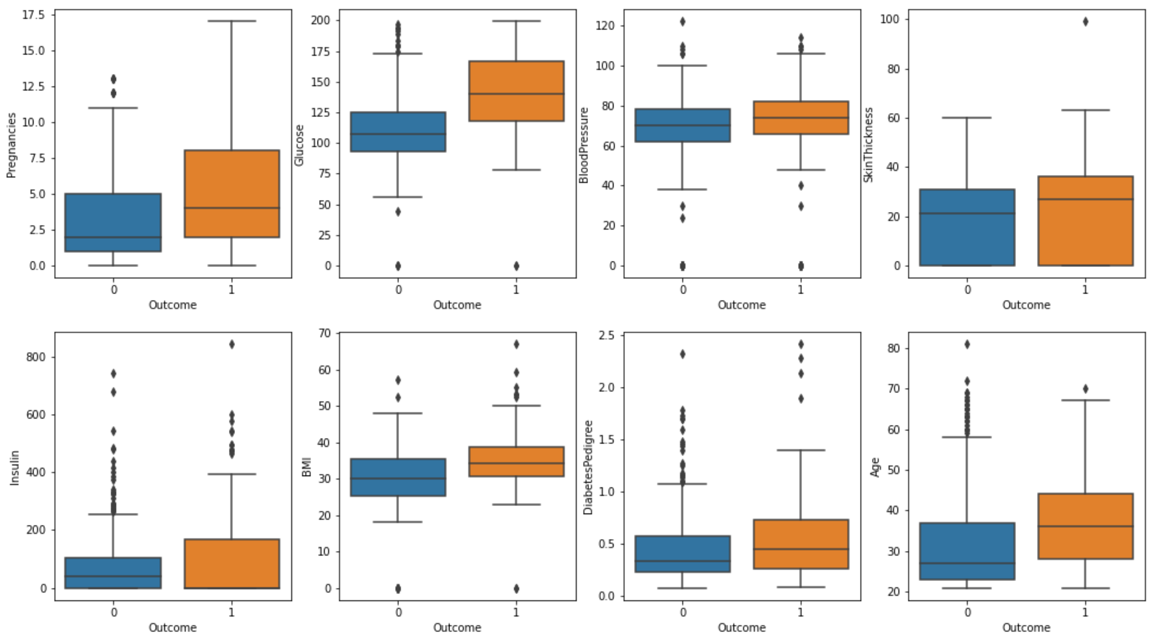

In [16]:

```
# each column somewhat resembles a normal distribution
# centering each of the columns around the mean would not be a bad idea here, but we
```

In [17]:

```
# boxplots for each column, showing how they change between women who do and who do 
plt.figure(figsize = (18,10))

plt.subplot(2,4,1)
sns.boxplot(x = diabetes['Outcome'], y = diabetes['Pregnancies'])

plt.subplot(2,4,2)
sns.boxplot(x = diabetes['Outcome'], y = diabetes['Glucose'])

plt.subplot(2,4,3)
sns.boxplot(x = diabetes['Outcome'], y = diabetes['BloodPressure'])

plt.subplot(2,4,4)
sns.boxplot(x = diabetes['Outcome'], y = diabetes['SkinThickness'])

plt.subplot(2,4,5)
sns.boxplot(x = diabetes['Outcome'], y = diabetes['Insulin'])

plt.subplot(2,4,6)
sns.boxplot(x = diabetes['Outcome'], y = diabetes['BMI'])

plt.subplot(2,4,7)
sns.boxplot(x = diabetes['Outcome'], y = diabetes['DiabetesPedigree'])
```

```
plt.subplot(2,4,8)
sns.boxplot(x = diabetes['Outcome'], y = diabetes['Age'])

# save the figure
plt.savefig("pythonImage3.jpg")
```




In [18]:
```
# BloodPressure and SkinThickness do not change as much as the others do
# they will be candidates for removal from the regression model later
```

In [19]:
```
# add the column 7 or more pregnancies
diabetes['SevenOrMorePregnancies'] = np.where(diabetes['Pregnancies'] >= 7, True, Fa
```

In [20]:
```
# save the diabetes dataset as a .csv and move over to R to carry on analysis
diabetes.to_csv("PimaDiabetes2.csv", index = False)
```

### 5.2 R Code

```
> #########
> # Student ID: 10724837
> # In this R file, I experiment with different regression models,
> # before choosing a final one and using it to predict the outcome
> # of some test data.
> #########
>
>
> # load and observe the data
> diabetes = read.csv("PimaDiabetes2.csv")
> attach(diabetes)
> dim(diabetes)
[1] 750  10
>
>
> # use a logistic regression model because Outcome is categorical with
> # two classes, '0' and '1'
> # fit a regression model with SevenOrMorePregnancies predicting Outcome
> model = glm(data = diabetes, family = binomial,
+             formula = Outcome ~ SevenOrMorePregnancies)
> summary(model)

Call:
glm(formula = Outcome ~ SevenOrMorePregnancies, family = binomial,
    data = diabetes)

Coefficients:
                           Estimate Std. Error z value Pr(>|z|)
(Intercept)                -0.91988    0.09151 -10.052  < 2e-16 ***
SevenOrMorePregnanciesTrue  1.18980    0.18224   6.529 6.63e-11 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

(Dispersion parameter for binomial family taken to be 1)

    Null deviance: 968.04  on 749  degrees of freedom
Residual deviance: 924.78  on 748  degrees of freedom
AIC: 928.78

Number of Fisher Scoring iterations: 4

>
>
>
```

```
> # try a regression model including all predictors, then apply the
> # idea of backward stepwise regression to improve
>
> # ignore SevenOrMorePregnancies and use Pregnancies instead
> model2 = lm(data = diabetes,
+             formula = Outcome ~ Pregnancies + Glucose + BloodPressure
+               + SkinThickness + Insulin + BMI + DiabetesPedigree
+                 + Age)
> summary(model2)

Call:
lm(formula = Outcome ~ Pregnancies + Glucose + BloodPressure +
    SkinThickness + Insulin + BMI + DiabetesPedigree + Age, data = diabetes)

Residuals:
    Min      1Q  Median      3Q     Max
-0.9901 -0.2977 -0.0988  0.3181  1.2319

Coefficients:
                   Estimate Std. Error t value Pr(>|t|)
(Intercept)      -0.8346708  0.0865735  -9.641  < 2e-16 ***
Pregnancies       0.0222857  0.0052112   4.277 2.15e-05 ***
Glucose           0.0058624  0.0005203  11.268  < 2e-16 ***
BloodPressure    -0.0023438  0.0008150  -2.876  0.00414 **
SkinThickness     0.0005035  0.0011312   0.445  0.65636
Insulin          -0.0001929  0.0001522  -1.268  0.20529
BMI               0.0129391  0.0021033   6.152 1.25e-09 ***
DiabetesPedigree  0.1388064  0.0456627   3.040  0.00245 **
Age               0.0022758  0.0015784   1.442  0.14977
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 0.4009 on 741 degrees of freedom
Multiple R-squared:  0.299,	Adjusted R-squared:  0.2914
F-statistic: 39.51 on 8 and 741 DF,  p-value: < 2.2e-16

>
>
> # SkinThickness is least significant, so try removing it
> model3 = lm(data = diabetes,
+             formula = Outcome ~ Pregnancies + Glucose + BloodPressure
+             + Insulin + BMI + DiabetesPedigree + Age)
> summary(model3)

Call:
lm(formula = Outcome ~ Pregnancies + Glucose + BloodPressure +
```

```
    Insulin + BMI + DiabetesPedigree + Age, data = diabetes)

Residuals:
     Min       1Q   Median       3Q      Max 
-1.00180 -0.29666 -0.09704  0.32055  1.23280 

Coefficients:
                   Estimate Std. Error t value Pr(>|t|)    
(Intercept)      -0.8343148  0.0865230  -9.643  < 2e-16 ***
Pregnancies       0.0222713  0.0052083   4.276 2.15e-05 ***
Glucose           0.0058268  0.0005138  11.340  < 2e-16 ***
BloodPressure    -0.0022905  0.0008057  -2.843  0.00459 ** 
Insulin          -0.0001657  0.0001393  -1.190  0.23452    
BMI               0.0132455  0.0019864   6.668 5.06e-11 ***
DiabetesPedigree  0.1409627  0.0453804   3.106  0.00197 ** 
Age               0.0022043  0.0015693   1.405  0.16057    
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Residual standard error: 0.4007 on 742 degrees of freedom
Multiple R-squared:  0.2988,	Adjusted R-squared:  0.2922 
F-statistic: 45.17 on 7 and 742 DF,  p-value: < 2.2e-16

> 
> 
> # adjusted R squared has increased, we can quite confidently remove
> # SkinThickness from the model permanently
> 
> # Insulin is next least significant, so try removing it
> model4 = lm(data = diabetes,
+             formula = Outcome ~ Pregnancies + Glucose + BloodPressure
+             + BMI + DiabetesPedigree + Age)
> summary(model4)

Call:
lm(formula = Outcome ~ Pregnancies + Glucose + BloodPressure + 
    BMI + DiabetesPedigree + Age, data = diabetes)

Residuals:
     Min       1Q   Median       3Q      Max 
-1.08142 -0.29786 -0.09529  0.31676  1.22349 

Coefficients:
                   Estimate Std. Error t value Pr(>|t|)    
(Intercept)      -0.8175909  0.0853974  -9.574  < 2e-16 ***
Pregnancies       0.0225821  0.0052032   4.340 1.62e-05 ***
```

```
Glucose           0.0056345  0.0004878  11.550  < 2e-16 ***
BloodPressure    -0.0023298  0.0008052  -2.893  0.00392 **
BMI               0.0130026  0.0019764   6.579 8.94e-11 ***
DiabetesPedigree  0.1333507  0.0449397   2.967  0.00310 **
Age               0.0023874  0.0015622   1.528  0.12689
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Residual standard error: 0.4008 on 743 degrees of freedom
Multiple R-squared:  0.2975,	Adjusted R-squared:  0.2918
F-statistic: 52.44 on 6 and 743 DF,  p-value: < 2.2e-16

>
>
> # adjusted R squared decreased, but only very slightly
> # likely worth its removal for the simplification of the model
>
> # Age is the next least significant, so try removing it
> model5 = lm(data = diabetes,
+             formula = Outcome ~ Pregnancies + Glucose + BloodPressure
+             + BMI + DiabetesPedigree)
> summary(model5)

Call:
lm(formula = Outcome ~ Pregnancies + Glucose + BloodPressure +
    BMI + DiabetesPedigree, data = diabetes)

Residuals:
     Min       1Q   Median       3Q      Max
-1.10385 -0.29599 -0.09924  0.31431  1.24685

Coefficients:
                   Estimate Std. Error t value Pr(>|t|)
(Intercept)      -0.7830064  0.0824180  -9.500  < 2e-16 ***
Pregnancies       0.0267621  0.0044302   6.041 2.42e-09 ***
Glucose           0.0057923  0.0004772  12.138  < 2e-16 ***
BloodPressure    -0.0021060  0.0007925  -2.657  0.00804 **
BMI               0.0127725  0.0019724   6.476 1.72e-10 ***
DiabetesPedigree  0.1362774  0.0449392   3.032  0.00251 **
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Residual standard error: 0.4011 on 744 degrees of freedom
Multiple R-squared:  0.2953,	Adjusted R-squared:  0.2905
F-statistic: 62.35 on 5 and 744 DF,  p-value: < 2.2e-16
```

```
>
>
> # adjusted R squared decreased, so keep Age for now
>
> # since Age and Pregnancies are correlated, try removing
> # Pregnancies instead and see the effect
> model6 = lm(data = diabetes,
+             formula = Outcome ~ Glucose + BloodPressure + BMI
+             + DiabetesPedigree + Age)
> summary(model6)

Call:
lm(formula = Outcome ~ Glucose + BloodPressure + BMI + DiabetesPedigree +
    Age, data = diabetes)

Residuals:
    Min      1Q  Median      3Q     Max
-1.0457 -0.2954 -0.1069  0.3258  1.2303

Coefficients:
                   Estimate Std. Error t value Pr(>|t|)
(Intercept)      -0.8469590  0.0861433  -9.832  < 2e-16 ***
Glucose           0.0056126  0.0004936  11.370  < 2e-16 ***
BloodPressure    -0.0022789  0.0008147  -2.797  0.00529 **
BMI               0.0130973  0.0019998   6.549 1.08e-10 ***
DiabetesPedigree  0.1208175  0.0453812   2.662  0.00793 **
Age               0.0059516  0.0013448   4.426 1.10e-05 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 0.4055 on 744 degrees of freedom
Multiple R-squared:  0.2797,	Adjusted R-squared:  0.2748
F-statistic: 57.77 on 5 and 744 DF,  p-value: < 2.2e-16

>
>
> # adjusted R squared decreased drastically, so we can quite
> # confidently keep Pregnancies in the model, and remove Age
>
> ###
> # At this point there are no other clear candidates for removal,
> # but we still have too many predictors. In our EDA, we saw that
> # BloodPressure had little effect on the Outcome, so try removing this
> ###
>
> model7 = lm(data = diabetes,
```

```
+             formula = Outcome ~ Pregnancies + Glucose + BMI +
+               DiabetesPedigree)
> summary(model7)

Call:
lm(formula = Outcome ~ Pregnancies + Glucose + BMI + DiabetesPedigree,
    data = diabetes)

Residuals:
    Min      1Q  Median      3Q     Max
-1.0923 -0.2944 -0.1037  0.3277  1.2262

Coefficients:
                   Estimate Std. Error t value Pr(>|t|)
(Intercept)      -0.8678511  0.0762912 -11.376  < 2e-16 ***
Pregnancies       0.0252114  0.0044094   5.718 1.56e-08 ***
Glucose           0.0056975  0.0004778  11.924  < 2e-16 ***
BMI               0.0114245  0.0019138   5.970 3.68e-09 ***
DiabetesPedigree  0.1363881  0.0451216   3.023  0.00259 **
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Residual standard error: 0.4028 on 745 degrees of freedom
Multiple R-squared:  0.2886,        Adjusted R-squared:  0.2848
F-statistic: 75.55 on 4 and 745 DF,  p-value: < 2.2e-16

>
>
> # although the adjusted R squared has dropped, the model is simpler
>
> # from here there is not much room for improvement, so this is our
> # final model
>
>
> # load the ToPredict file
> ToPredict = read.csv("ToPredict.csv")
>
> # use our final model to make our predictions!
> predict(model7, ToPredict)
        1         2         3         4         5
0.5255120 0.3279125 0.1505665 0.6883586 0.6418316
```