Principles of Data Science

Task 1 – Group Portfolio

Avwerosuo Godstime Emekeme

Christopher Eyare Eban

Ndubuaku George Ekwueme

Tawana Joseph Mhishi

grouptask.R

QUESTION 1

```
# locking a sample
set.seed(341)

library(tidyverse)

## ── Attaching core tidyverse packages ──────────────────────── tidyverse
2.0.0 ──
## ✓ dplyr     1.1.3     ✓ readr     2.1.4
## ✓ forcats   1.0.0     ✓ stringr   1.5.0
## ✓ ggplot2   3.4.3     ✓ tibble    3.2.1
## ✓ lubridate 1.9.2     ✓ tidyr     1.3.0
## ✓ purrr     1.0.2
## ── Conflicts ────────────────────────────────────────
tidyverse_conflicts() ──
## ✖ dplyr::filter() masks stats::filter()
## ✖ dplyr::lag()    masks stats::lag()
## ℹ Use the conflicted package (<http://conflicted.r-lib.org/>) to force all
conflicts to become errors

# Read Pima Indians Diabetes Dataset as csv file

diabetesdata <- read.csv('Pima_Indians_Diabetes_Dataset1.csv')

# To take stratified random samples

strata_sample = diabetesdata %>%

  group_by(Target) %>%

  slice_sample(n=200) %>%

  ungroup()

View(strata_sample)

# Select you own filename instead of 'your_dataset.csv' for your sample

write_csv(strata_sample, 'portfoliodata.csv')
#strata_sample
read_csv('portfoliodata.csv')

## Rows: 400 Columns: 9
## ── Column specification
────────────────────────────────────────────────────────

## Delimiter: ","
```

```
## chr (1): Target
## dbl (8): Number.of.times.pregnant, Plasma.glucose.concentration,
Diastolic.b...
##
## ℹ Use `spec()` to retrieve the full column specification for this data.
## ℹ Specify the column types or set `show_col_types = FALSE` to quiet this
message.

## # A tibble: 400 × 9
##    Number.of.times.pregnant Plasma.glucose.concentration
Diastolic.blood.press…¹
##                       <dbl>                       <dbl>
<dbl>
##  1                        0                         123
72
##  2                        5                         112
66
##  3                        6                         124
72
##  4                        9                         164
84
##  5                        0                         131
88
##  6                        2                         146
70
##  7                        4                         115
72
##  8                        0                         162
76
##  9                        1                         168
88
## 10                        8                         100
74
## # ℹ 390 more rows
## # ℹ abbreviated name: ¹Diastolic.blood.pressure
## # ℹ 6 more variables: Triceps.skinfold.thickness <dbl>,
## #   X2.Hour.serum.insulin <dbl>, Body.mass.index <dbl>,
## #   Diabetes.pedigree.function <dbl>, Age <dbl>, Target <chr>

# PCA biplot
diabetesdata <- read_csv('portfoliodata.csv')

## Rows: 400 Columns: 9
## ── Column specification
──────────────────────────────────────────────────────

## Delimiter: ","
## chr (1): Target
## dbl (8): Number.of.times.pregnant, Plasma.glucose.concentration,
Diastolic.b...
##
```

```
## ℹ Use `spec()` to retrieve the full column specification for this data.
## ℹ Specify the column types or set `show_col_types = FALSE` to quiet this
message.

diabetes_data_bi = select(diabetesdata,-Target)

pca_diabetesdata = prcomp(diabetes_data_bi, scale.=TRUE)

summary(pca_diabetesdata)

## Importance of components:
##                           PC1    PC2    PC3    PC4    PC5    PC6     PC7
## Standard deviation     1.4243 1.2940 1.0039 0.9525 0.9230 0.8602 0.65950
## Proportion of Variance 0.2536 0.2093 0.1260 0.1134 0.1065 0.0925 0.05437
## Cumulative Proportion  0.2536 0.4629 0.5888 0.7023 0.8087 0.9012 0.95562
##                            PC8
## Standard deviation     0.59585
## Proportion of Variance 0.04438
## Cumulative Proportion  1.00000
```

Note that PC1 == 25.36% while PC2 == 20.93%, and the sum of both PC1 and PC2 == 46.29% variance.

```
# Biplot

#install.packages("ggfortify")

library(ggfortify)

autoplot(pca_diabetesdata,

         label=TRUE, label.size=3, shape=FALSE,

         loadings=TRUE, loadings.label=TRUE,

         data=diabetesdata,  col = 'Target')
```

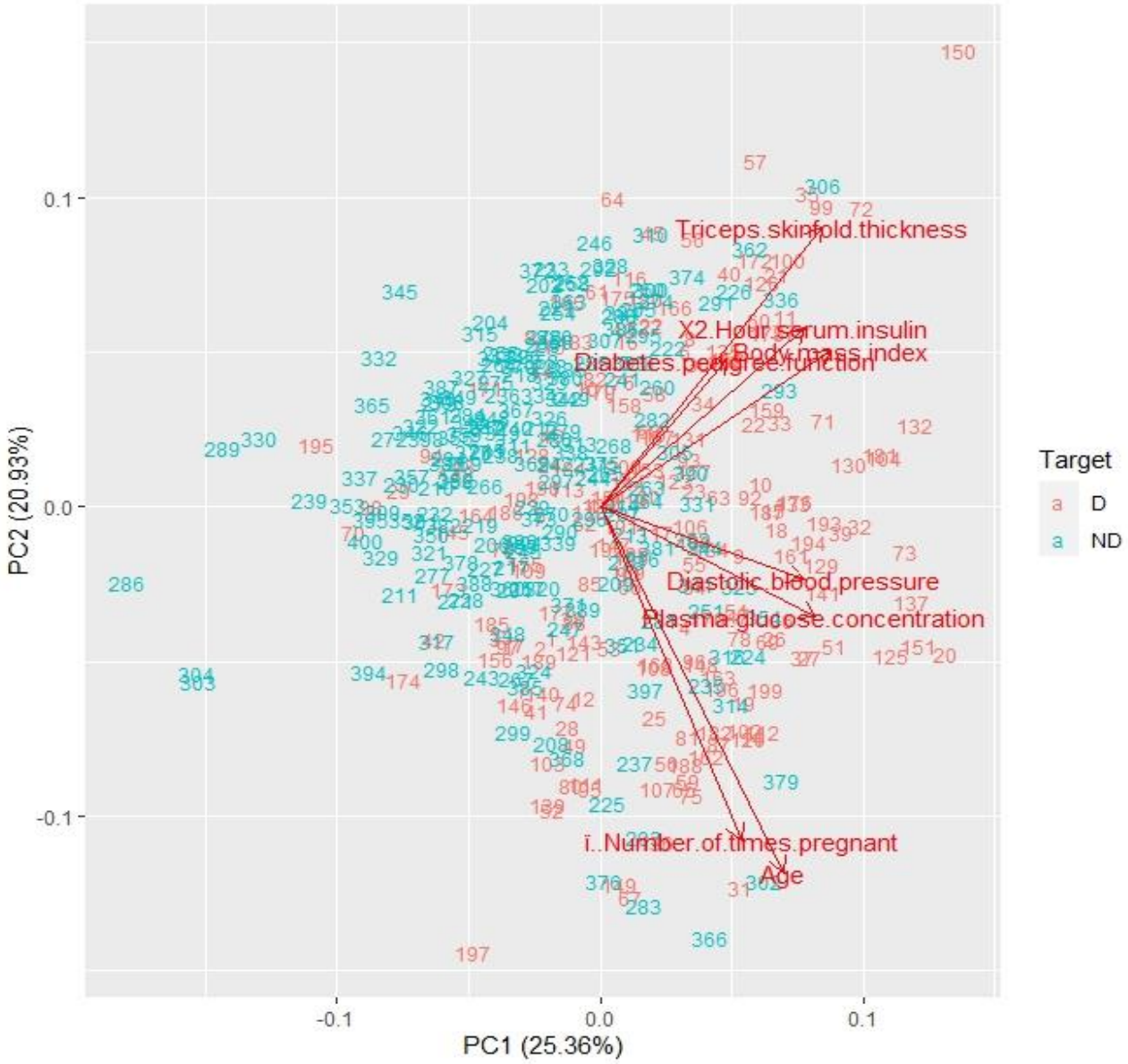

Figure 1: Biplot of PC1 and PC2

Analysis from Figure 1 depicts that all the 8 variables are pointing in the same direction, the right. Even more specifically, a strong correlation between three groups of variables.

Group 1: Triceps skinfold thickness, X2 Hour serum insulin, Body mass index and diabetes pedigree function.

Group 2: Diastolic blood pressure and plasma glucose concentration.

Group 3: Number of times pregnant and Age.

The above groupings represent the combination of variables that are strongly correlated to themselves as seen in Figure 1. it is relevant to note that Figure 1 has a sum of 46.29% variance from PC1 == 25.36% and PC2 == 20.93%. However, both PC1 and PC2 seem insufficient for it is a rule of thumb to keep at least 80% of the explained variance (Kaloyanova, 2021).

```
# Screeplot

diabetesdata_var =
100*((pca_diabetesdata$sd)^2)/(sum((pca_diabetesdata$sd)^2))

diabetesdata_var

## [1] 25.356143 20.931036 12.597288 11.341096 10.649621  9.250013  5.436785
## [8]  4.438018

cumsum(diabetesdata_var)

## [1]  25.35614  46.28718  58.88447  70.22556  80.87518  90.12520  95.56198
## [8] 100.00000

ggplot(NULL,aes(x=1:8,y=diabetesdata_var)) +

  geom_col() +

  ggtitle('Scree Plot Pima Indians Diabetes') +

  xlab('Principal Component (PC)') +

  ylab('Percentage Variance ') +

  geom_point() +

  geom_line()
```

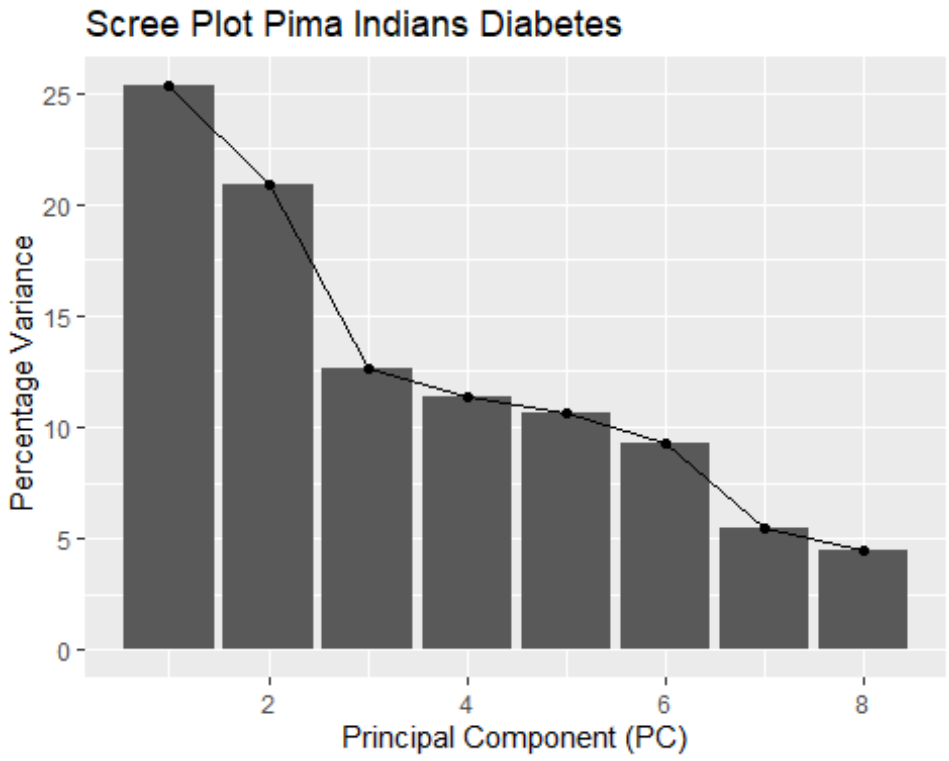

Figure 2: Scree Plot

Note that PC3 is 12.60%.

Note that the scree plot shows the variance starts to form a straight line from the third variable or the third Principal component. Sum of both PC1, PC2, and PC3 == 58.89% which by our analysis and interpretation is generally insufficient, as a sufficient variance is one that is at least 80%.

An analysis of figure 2 shows that PC3 has a 12.60% variance, while the visualization of figure 3 depicts that the variance starts to form a straight line from PC3. PC3 being a 12.60% brings the sum of PC1, PC2, and PC3 to 58.89% which can loosely be interpreted as generally insignificant still. (Kaloyanova, 2021).

```
# Loadings plot PC1-PC3

diabetes_loadings = as.data.frame(pca_diabetesdata$rotation[,1:3])
diabetes_loadings
```

```
##                                 PC1        PC2          PC3
## Number.of.times.pregnant     0.2566981 -0.5158671  0.033850736
## Plasma.glucose.concentration 0.3922446 -0.1708204 -0.469153224
## Diastolic.blood.pressure     0.3706713 -0.1122288  0.466093270
## Triceps.skinfold.thickness   0.4024848  0.4329463  0.055285899
## X2.Hour.serum.insulin        0.3725939  0.2755799 -0.611407695
## Body.mass.index              0.4198711  0.2407423  0.374385391
```

```
## Diabetes.pedigree.function   0.2308880  0.2271402  0.210761687
## Age                          0.3353084 -0.5649512  0.006755271

diabetes_loadings$Medical_Data = row.names(diabetes_loadings)

diabetes_loadings = gather(diabetes_loadings, key='Component',
value='Diabetes.Pedigree.Function', -Medical_Data)

ggplot(diabetes_loadings, aes(x=Medical_Data,y=Diabetes.Pedigree.Function)) +

  geom_bar(stat='identity') +

  facet_grid(Component~.) +

  ggtitle('Loadings for Principle Components')
```

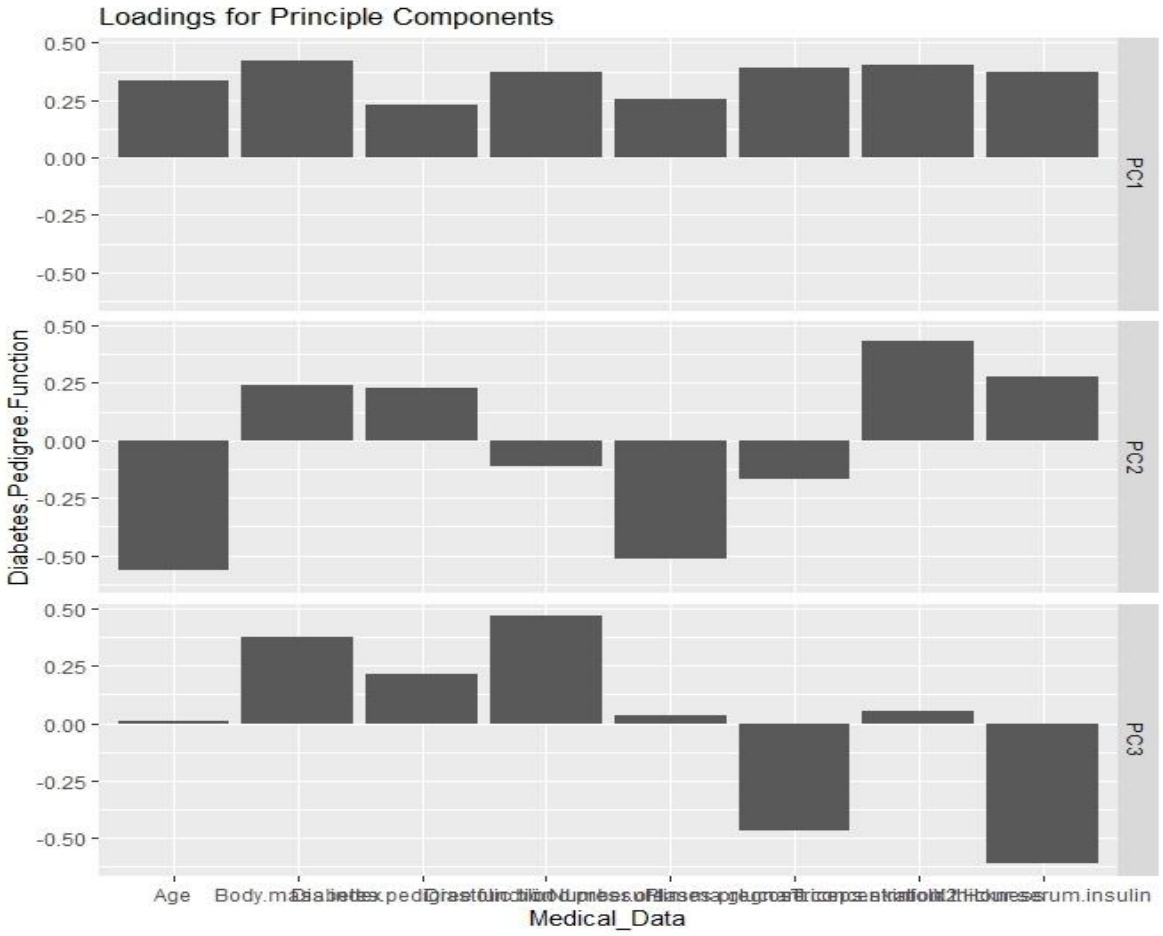

Figure 3

Although our Loadings plot is for PC1 - PC8, we shall only consider PC1 - PC3 as underlined in question number 1(a), Second paragraph.

Note: 1. PC1 has ALL 8 variables on positive, with "BMI" having the highest value, followed by both "Triceps skinfold thickness", "plasma Glucose concentration", and "X2 hour serum insulin".

2. PC2 has 4 positive variables - "BMI", "Diabetes pedigree function", "Triceps skinfold thickness", and "X2. Hour serum insulin". and 4 negative variables - "Age", "Diastolic blood pressure", "Number of times pregnant", and "plasma Glucose concentration".

3. PC3 has 6 positive variables - "Age", "BMI", Diabetes pedigree function", "Diastolic blood pressure", "Number of times pregnant", and "Triceps skinfold thickness", and 2 negative variables - "plasma Glucose concentration", and X2. Hour serum insulin".

Figure 3 analysis shall be focused on PC1, PC2, and PC3 only, as per the instructions given. PC1 is seen here to all be in positive, which can be interpreted to mean all the variables are positively associated with PC1. The highest value being body mass index with 0.4198. While PC2 is seen here to have 4 positive variables and 4 negative variables. The highest positive value being Triceps skinfold thickness with 0.4329, and Age as the highest negative value with –0.5649. Consequently, this means that PC2 has 4 variables (negatives) not captured. Again, PC3 appears to have 6 positive variables and 2 negative variables with Diastolic blood pressure with 0.4660, leading the positives and X2 Hour serum insulin leading the negative correlated variables with –0.6114.

```
#biplot for pc2 and pc3

autoplot(pca_diabetesdata,x=2,y=3,

         label=TRUE, label.size=3, shape=FALSE,

         loadings=TRUE, loadings.label=TRUE,

         data=diabetesdata, colour='Target')
```

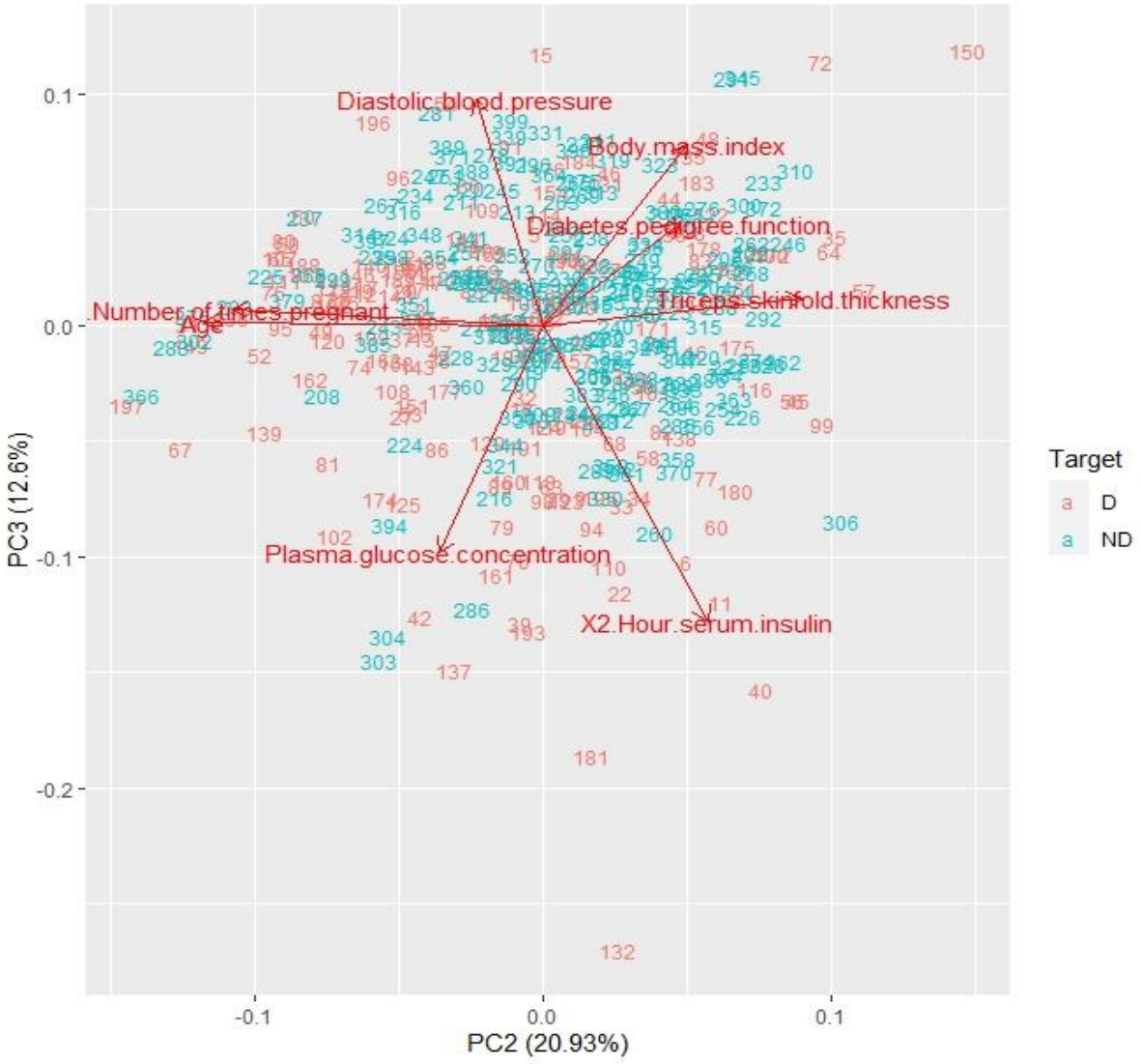

Figure 4: Biplot of PC2 and PC3

```
#INTERPRETATION

Note that PC3 == 12.60%, and PC2 == 20.93%,

Remember that the sum of PC1 and PC2 as interpreted in Question 1(a) ==
46.29% variance.

Consequently, the sum of [PC1, PC2] and PC3 == 58.89% variance.

#biplot for pc3 and pc4
```

Figure 4 illustrates Triceps skinfold thickness, X2 Hour serum insulin, BMI, and Diabetes pedigree function to all be in positive loadings, particularly Triceps skinfold thickness and X2 hour serum insulin. While PC3 has 6 variables on positive which is the same as the analysis in Figure 3. Again, it is relevant to note that X2 hour serum insulin is the variable with the most negative influence here.

Figure 4 depicts a combine sum of 33.53% variance, that is PC3 == 12.60% and PC2 == 20.93%

```
#autoplot(pca_diabetesdata,x=3,y=4,
#         label=TRUE, label.size=3, shape=FALSE,
#         loadings=TRUE, loadings.label=TRUE,
#         data=diabetesdata, colour='Target')
```

QUESTION 2

```
#Factor Analysis
#install.packages("psych")
#install.packages("factoextra")
#install.packages("nFactors")

library(psych)

##
## Attaching package: 'psych'
##
## The following objects are masked from 'package:ggplot2':
##
##     %+%, alpha

library(factoextra)

## Welcome! Want to learn more? See two factoextra-related books at
https://goo.gl/ve3WBa

library(nFactors)

## Loading required package: lattice
##
## Attaching package: 'nFactors'
##
## The following object is masked from 'package:lattice':
##
##     parallel

cor(diabetes_data_bi)

##                               Number.of.times.pregnant
## Number.of.times.pregnant                   1.000000000
## Plasma.glucose.concentration               0.147471458
## Diastolic.blood.pressure                   0.109717482
## Triceps.skinfold.thickness                 0.006234575
## X2.Hour.serum.insulin                     -0.029398734
## Body.mass.index                            0.010066798
## Diabetes.pedigree.function                -0.034912069
## Age                                        0.530187487
##                               Plasma.glucose.concentration
## Number.of.times.pregnant                        0.14747146
## Plasma.glucose.concentration                    1.00000000
## Diastolic.blood.pressure                        0.14792826
## Triceps.skinfold.thickness                      0.02316757
## X2.Hour.serum.insulin                           0.27490488
```

```
## Body.mass.index                                0.19050662
## Diabetes.pedigree.function                     0.07513530
## Age                                            0.30313368
##                              Diastolic.blood.pressure
## Number.of.times.pregnant                   0.10971748
## Plasma.glucose.concentration               0.14792826
## Diastolic.blood.pressure                   1.00000000
## Triceps.skinfold.thickness                 0.16725289
## X2.Hour.serum.insulin                      0.06735879
## Body.mass.index                            0.23245556
## Diabetes.pedigree.function                 0.05874373
## Age                                        0.27277894
##                              Triceps.skinfold.thickness
X2.Hour.serum.insulin
## Number.of.times.pregnant                    0.006234575           -
0.02939873
## Plasma.glucose.concentration                0.023167567
0.27490488
## Diastolic.blood.pressure                    0.167252887
0.06735879
## Triceps.skinfold.thickness                  1.000000000
0.42602648
## X2.Hour.serum.insulin                       0.426026483
1.00000000
## Body.mass.index                             0.409542642
0.12574752
## Diabetes.pedigree.function                  0.194405352
0.10918208
## Age                                        -0.102315666
0.02529774
##                              Body.mass.index Diabetes.pedigree.function
## Number.of.times.pregnant           0.0100668               -0.034912069
## Plasma.glucose.concentration       0.1905066                0.075135299
## Diastolic.blood.pressure           0.2324556                0.058743728
## Triceps.skinfold.thickness         0.4095426                0.194405352
## X2.Hour.serum.insulin              0.1257475                0.109182076
## Body.mass.index                    1.0000000                0.178075859
## Diabetes.pedigree.function         0.1780759                1.000000000
## Age                                0.0463958                0.004436249
##                                         Age
## Number.of.times.pregnant       0.530187487
## Plasma.glucose.concentration   0.303133676
## Diastolic.blood.pressure       0.272778935
## Triceps.skinfold.thickness    -0.102315666
## X2.Hour.serum.insulin          0.025297736
## Body.mass.index                0.046395796
## Diabetes.pedigree.function     0.004436249
## Age                            1.000000000
```

```
# strongest correlation is between Number of times Pregnant and Age which is
0.530*** as compared to the other combinations.

# Followed by insulin and TST which is 0.426***

#getting the eigen values of the sample dataset


ev=eigen(cor(diabetes_data_bi))

Ns=nScree(x=ev$values)

plotnScree(Ns, legend = F)
```

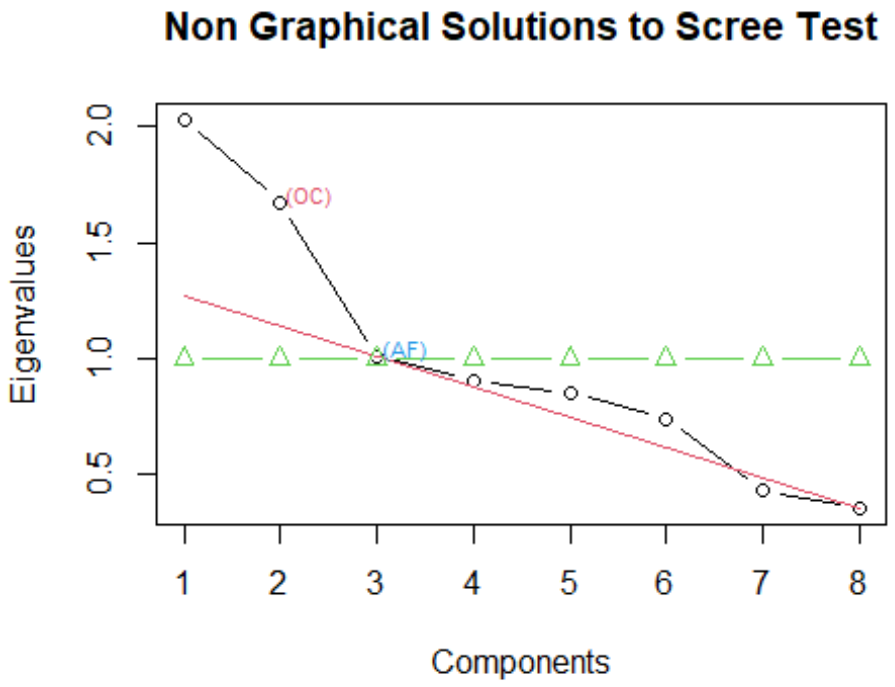

Figure 5: nScree plot

```
print(ev$values)

## [1] 2.0284914 1.6744829 1.0077831 0.9072876 0.8519697 0.7400010 0.4349428
## [8] 0.3550414

# Eigen values for all 8 components:

# 2.0284914 1.6744829 1.0077831 0.9072876 0.8519697 0.7400010 0.4349428
0.3550414

#next this shows us that there are 3 significant factors.
```

```r
fit1=factanal(diabetes_data_bi,3, rotation='none')

print(fit1,digits=3,cutoff=0, sort=T)
```

```
##
## Call:
## factanal(x = diabetes_data_bi, factors = 3, rotation = "none")
##
## Uniquenesses:
##     Number.of.times.pregnant Plasma.glucose.concentration
##                        0.716                        0.821
##     Diastolic.blood.pressure   Triceps.skinfold.thickness
##                        0.833                        0.569
##        X2.Hour.serum.insulin              Body.mass.index
##                        0.005                        0.428
##   Diabetes.pedigree.function                          Age
##                        0.932                        0.005
##
## Loadings:
##                               Factor1 Factor2 Factor3
## X2.Hour.serum.insulin          0.709  -0.702  -0.003
## Age                            0.719   0.691  -0.001
## Body.mass.index                0.122  -0.059   0.744
## Number.of.times.pregnant       0.354   0.399   0.016
## Plasma.glucose.concentration   0.405   0.017   0.119
## Diastolic.blood.pressure       0.240   0.145   0.298
## Triceps.skinfold.thickness     0.225  -0.382   0.484
## Diabetes.pedigree.function     0.080  -0.076   0.237
##
##                Factor1 Factor2 Factor3
## SS loadings      1.439   1.305   0.947
## Proportion Var   0.180   0.163   0.118
## Cumulative Var   0.180   0.343   0.461
##
## Test of the hypothesis that 3 factors are sufficient.
## The chi square statistic is 25.66 on 7 degrees of freedom.
## The p-value is 0.00058
```

```r
#next we get the 3 factors and their chi square stat and also p-value

fit2=factanal(diabetes_data_bi,3, rotation='varimax')

print(fit2,digits=3,cutoff=0, sort=T)
```

```
##
## Call:
## factanal(x = diabetes_data_bi, factors = 3, rotation = "varimax")
##
```

```
## Uniquenesses:
##     Number.of.times.pregnant Plasma.glucose.concentration
##                        0.716                        0.821
##     Diastolic.blood.pressure   Triceps.skinfold.thickness
##                        0.833                        0.569
##        X2.Hour.serum.insulin              Body.mass.index
##                        0.005                        0.428
##   Diabetes.pedigree.function                          Age
##                        0.932                        0.005
##
## Loadings:
##                               Factor1 Factor2 Factor3
## Number.of.times.pregnant       0.532  -0.031  -0.010
## Age                            0.997   0.026  -0.036
## X2.Hour.serum.insulin          0.006   0.984   0.166
## Triceps.skinfold.thickness    -0.091   0.340   0.554
## Body.mass.index                0.074   0.001   0.753
## Plasma.glucose.concentration   0.303   0.252   0.152
## Diastolic.blood.pressure       0.284   0.017   0.294
## Diabetes.pedigree.function     0.012   0.069   0.252
##
##                 Factor1 Factor2 Factor3
## SS loadings       1.463   1.153   1.075
## Proportion Var    0.183   0.144   0.134
## Cumulative Var    0.183   0.327   0.461
##
## Test of the hypothesis that 3 factors are sufficient.
## The chi square statistic is 25.66 on 7 degrees of freedom.
## The p-value is 0.00058
```

```
fit = fa(diabetes_data_bi, nfactors=3, cutoff=0.5,rotate='varimax')
```

```
## Warning in fa.stats(r = r, f = f, phi = phi, n.obs = n.obs, np.obs =
np.obs, :
## The estimated weights for the factor scores are probably incorrect.  Try a
## different factor score estimation method.
```

```
## Warning in fac(r = r, nfactors = nfactors, n.obs = n.obs, rotate = rotate,
: An
## ultra-Heywood case was detected.  Examine the results carefully
```

```
library(psych)
psych::smc(diabetes_data_bi)
```

```
##     Number.of.times.pregnant Plasma.glucose.concentration
##                   0.29721933                   0.20617463
##     Diastolic.blood.pressure   Triceps.skinfold.thickness
##                   0.14021021                   0.37169901
##        X2.Hour.serum.insulin              Body.mass.index
```

```
##                     0.27128701                     0.24061669
##   Diabetes.pedigree.function                            Age
##                     0.05494387                     0.39056939

# The communalities tells us how well each variable is explained by the
factors.
# the closer the communality is to 1 the better the variable is explained

#Getting the Kaiser-Meyer-olkin measure
KMO(diabetes_data_bi)

## Kaiser-Meyer-Olkin factor adequacy
## Call: KMO(r = diabetes_data_bi)
## Overall MSA =  0.53
## MSA for each item =
##     Number.of.times.pregnant Plasma.glucose.concentration
##                         0.51                         0.55
##     Diastolic.blood.pressure   Triceps.skinfold.thickness
##                         0.65                         0.49
##        X2.Hour.serum.insulin              Body.mass.index
##                         0.48                         0.57
##   Diabetes.pedigree.function                          Age
##                         0.76                         0.53

fa.diagram(fit)
```

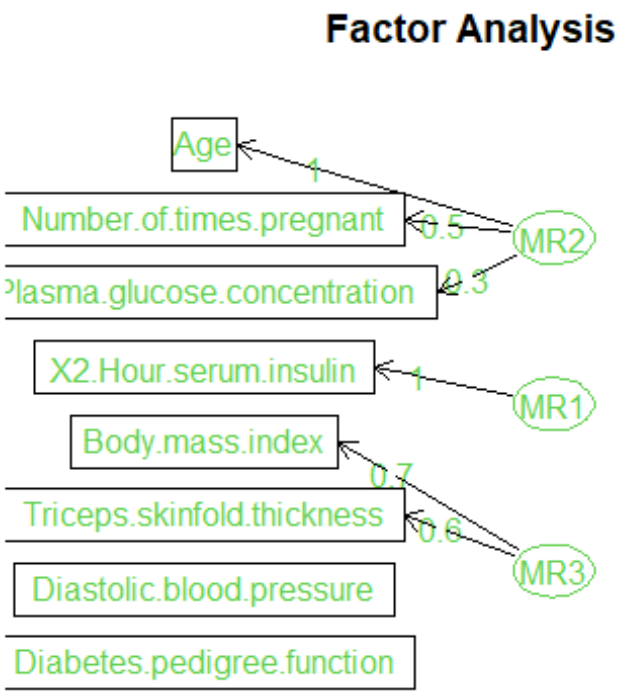

Figure 6: Factor Analysis

QUESTION 3

```
library(cluster)
BBB = scale(diabetes_data_bi)
D = dist(BBB, method="euclidean")
cluster_results = agnes(D, method='complete')
# "average", "single", "complete", "ward" these are the different hierarchial
clustering methods
plot(cluster_results, which.plots=2,
     main='Cluster of Observation of 8 variables (Euclidean/Ward)', ylab =
"Distance")
rect.hclust(cluster_results, k=4, border=2)
```

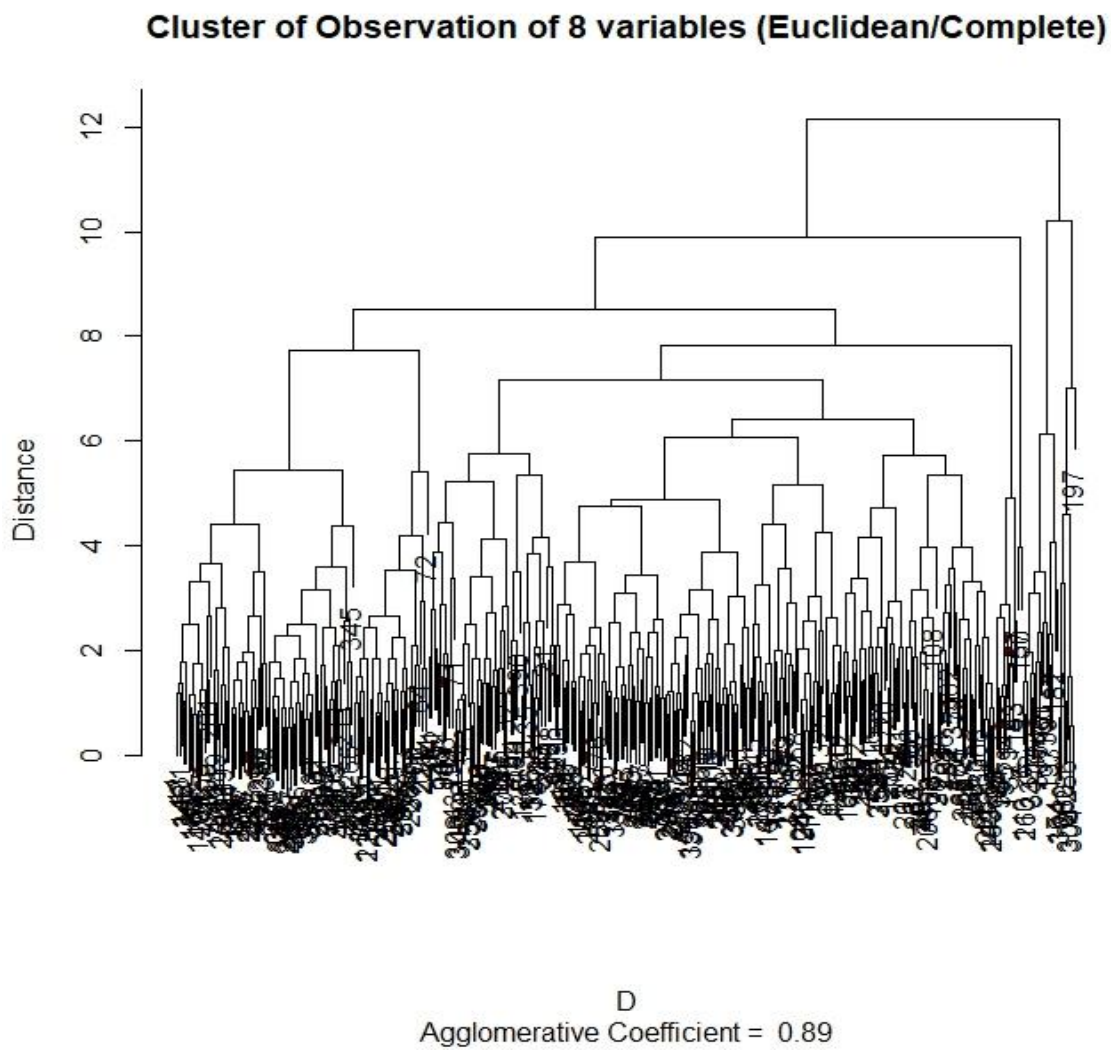

Figure 7: Euclidean/Complete Dendrogram

```
clusters = cutree(cluster_results, k=4)
```

In the above code, "Euclidean" distance metrics and a "complete" hierarchical clustering method was used to carry out the analysis

The distance metrics is to calculate the distance between each data points. The hierarchical clustering method in this case 'complete" method is a way of grouping these individual data points into clusters based on their distance. Having an agglomerative coefficient of 0.89, the dendrogram explains how each individual data point is closely related to each other.

The higher the agglomerative coefficient, the better the data points are good at forming clusters

Note: Agglomerative clustering method starts from bottom to top, the closer they are to the bottom the higher their similarities.

```
#         3B
BBB = scale(diabetes_data_bi)
D = dist(t(BBB), method="manhattan")
cluster_results = agnes(D, method='ward')
plot(cluster_results, which.plots=2,
     main='Cluster of Observation of 8 variables (manhattan/Ward)', ylab =
"Distance")
```

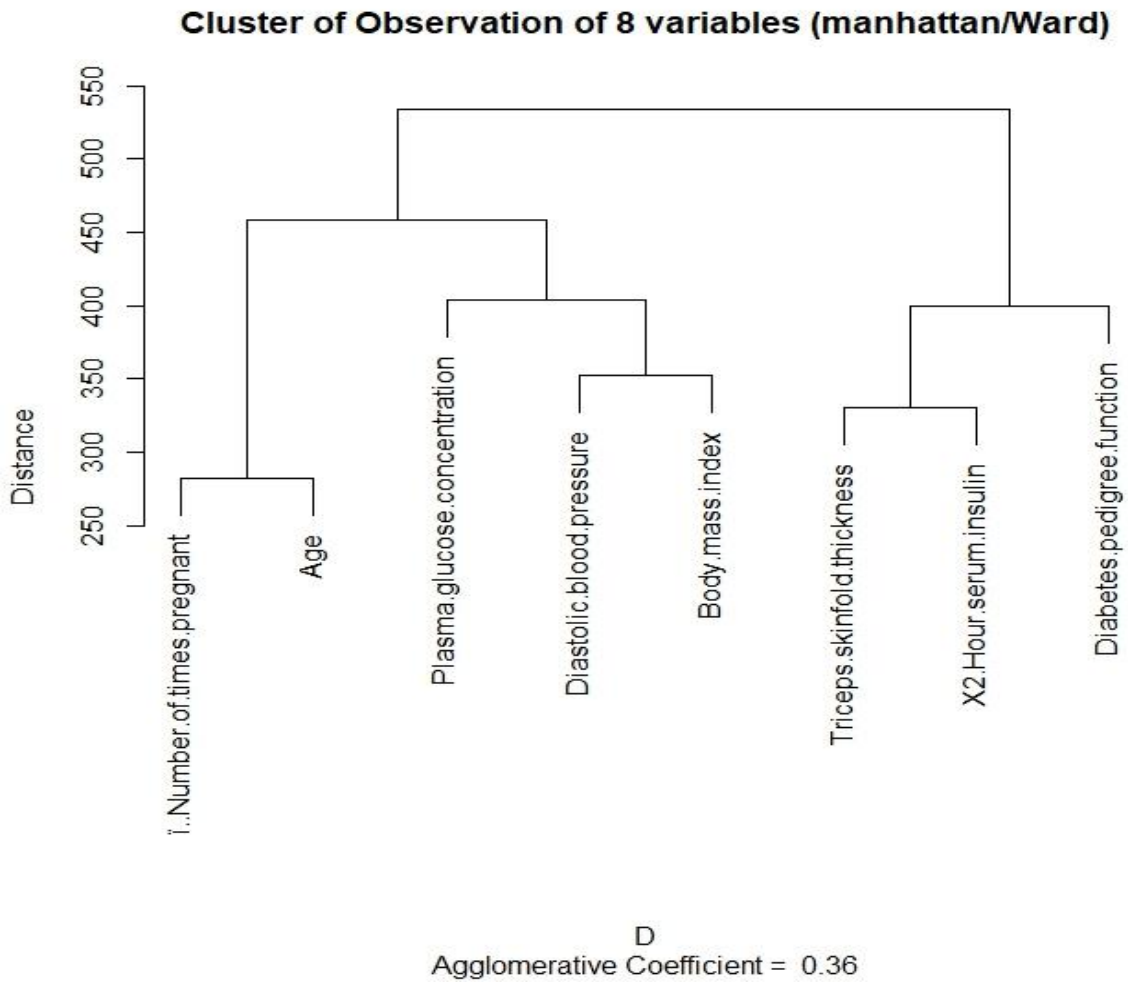

Figure 8: Manhattan/ward dendrogram

```
#rect.hclust(cluster_results, k=5, border=3)
clusters = cutree(cluster_results, k=5)
```

From this dendrogram the 'Number of times pregnant' and 'age' were the first to merge making them closely related to each other. This means they are similar and belong to the same group

Next was "triceps skinfold thickness" and "X2 hour serum insulin" making them also related to each other the distance metrics is important in explaining how each variable is similar with the other. The higher the distance the less similar it is.

For example, the distance between the cluster of number of times pregnant/age and the cluster of Plasma Glucose Concentration/Diastolic Blood Pressure/Body Mass Index and

BMI is much approximately 450 - 280 = 170 making them not as related due to the distance metrics being 170 over 50% of the whole distance in the distance metrics

Also, the agglomerative coefficient is 0.36 this coefficient is low and therefore suggest that the data points are more spread out and not forming clear clusters due to lack of similarities or correlation.

QUESTION 4

- One thing that the PCA (Principal Component Analysis), factor analysis and cluster analysis have in common is that they are all not significant enough to explain the variance of the dataset. Firstly, the PCA shows that both PC1 and PC2 seem insufficient for it is a rule of thumb to keep at least 80% of the explained variance. (Kaloyanova, 2021). Secondly, the factor analysis shows that the number of factors needed to explain 80% of the variance is 5 whereas the scree test shows that we can only retain 3 factors in the analysis. Thirdly, the cluster analysis shows that the agglomerative coefficient was too low (0.36) which shows that the clusters lack similarities with each other.

- Another issue that the PCA and the factor analysis have is common in this dataset is that the PCA shows that more than 4 principal components are needed to explain for 80% of the variance and this is shown by the scree plot. PC1 – 25.36%, PC2 – 20.93%, PC3-12.6%, PC4 – 11.34%, PC5 – 10.65% = 80.87% variance. The factor analysis shows that more than 3 factors are needed to explain the variance for the dataset, but the cumulative variance falls short of the required which is 80% .

Group Member Contributions to the Task

The group task was completed through a collaborative effort, with each member contributing to all aspects of the work while also taking primary responsibility for specific questions.

- **Christopher Eyare Eban** led the analysis for **Question 1**, which involved **Principal Component Analysis (PCA)**, generating biplots, scree plots, and loadings, as well as interpreting the results. His work provided the foundation for understanding the key components and their contributions to the dataset.
- **Tawana Joseph Mhishi** and **Avwerosuo Godstime Emekeme** jointly conducted the **Factor Analysis** for **Question 2**, sharing ideas and working together to compute loadings, communalities, and interpret the factors extracted. Their teamwork ensured a thorough and accurate analysis.
- **Ndubuaku George Ekwueme** took the lead on **Question 3**, performing the **Cluster Analysis** to group observations and identify patterns within the dataset. His analysis was instrumental in revealing relationships among the variables and participants.
- All four group members actively contributed to **Question 4**, which involved interpreting the overall results, critically assessing the analyses conducted, and drafting the final report. The collaborative effort ensured a cohesive and well-documented portfolio that explained the methods used, the results obtained, and their interpretations.

In summary, while each member had a specific focus area, the group worked together throughout the project, providing input, support, and feedback for each section to ensure high-quality outcomes and a comprehensive analysis.

REFERENCES

Koch, I. (2013). *Principal Component Analysis*. In *Analysis of Multivariate and High-Dimensional Data* (Cambridge Series in Statistical and Probabilistic Mathematics. Cambridge University Press.

Denis, D. J. (2020). *Univariate, bivariate, and multivariate statistics using r : Quantitative tools for data analysis and data science*. John Wiley & Sons, Incorporated.