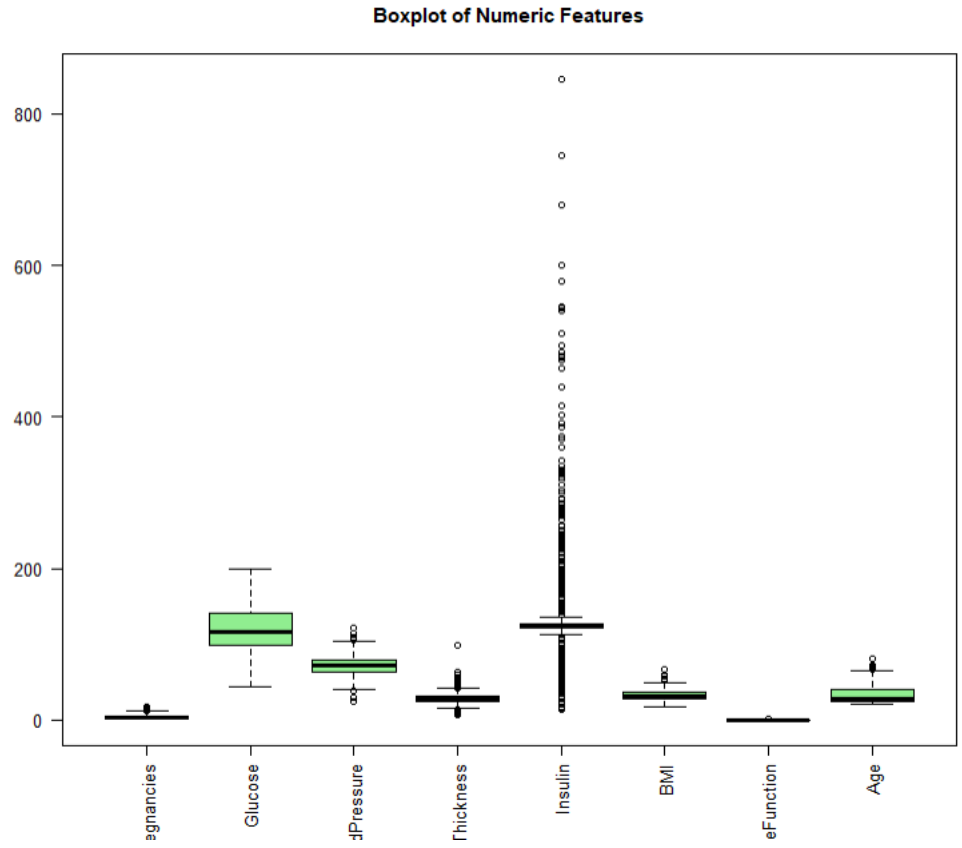
Project work 3
Insights from Data
Viivi Väisänen

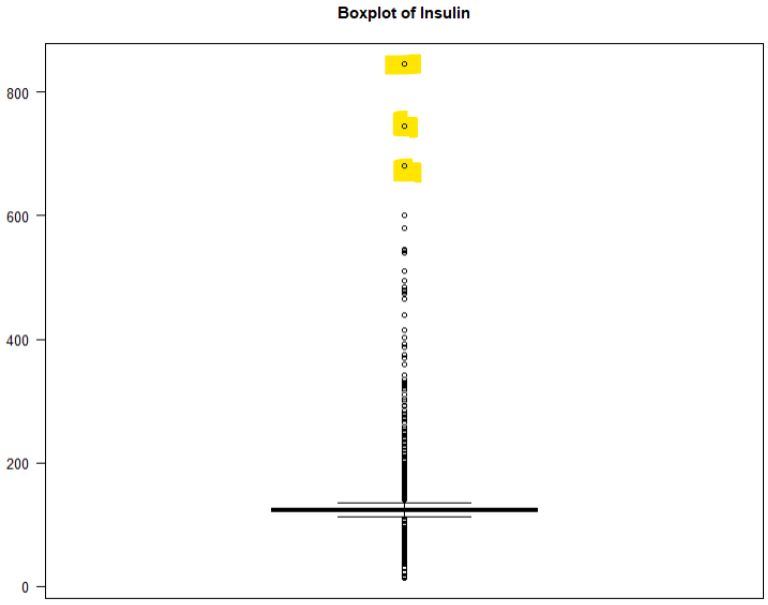
Data preprocessing

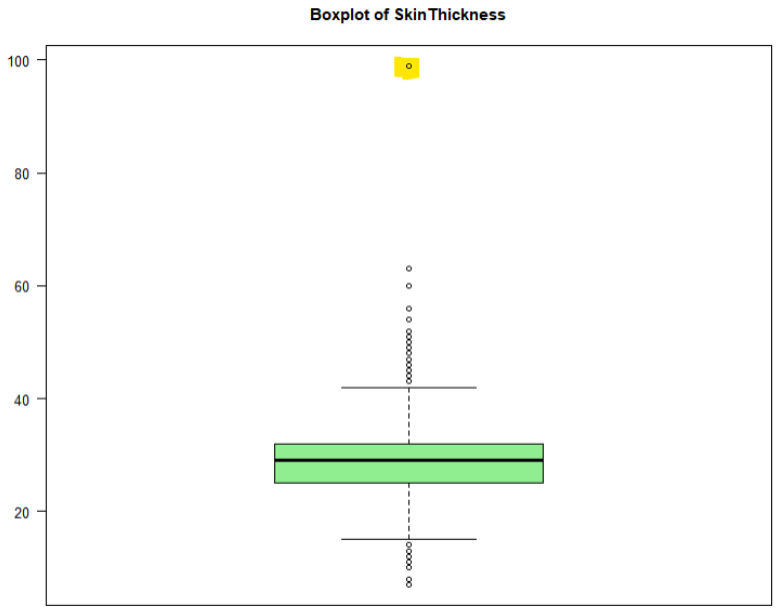
1. Handling invalid values
   I visualized the amount of zero values in the dataset, and replaced 0 values with the median of the respective values. I also converted Outcome column into categorical type (factor.)

2. Outliers
   I visualized the outliers with boxplot:



Columns *SkinThickness* and *Insulin* show few extreme outliers that can be seen below. I removed those few datapoints from the dataset, so that they dot not interfere with clustering.

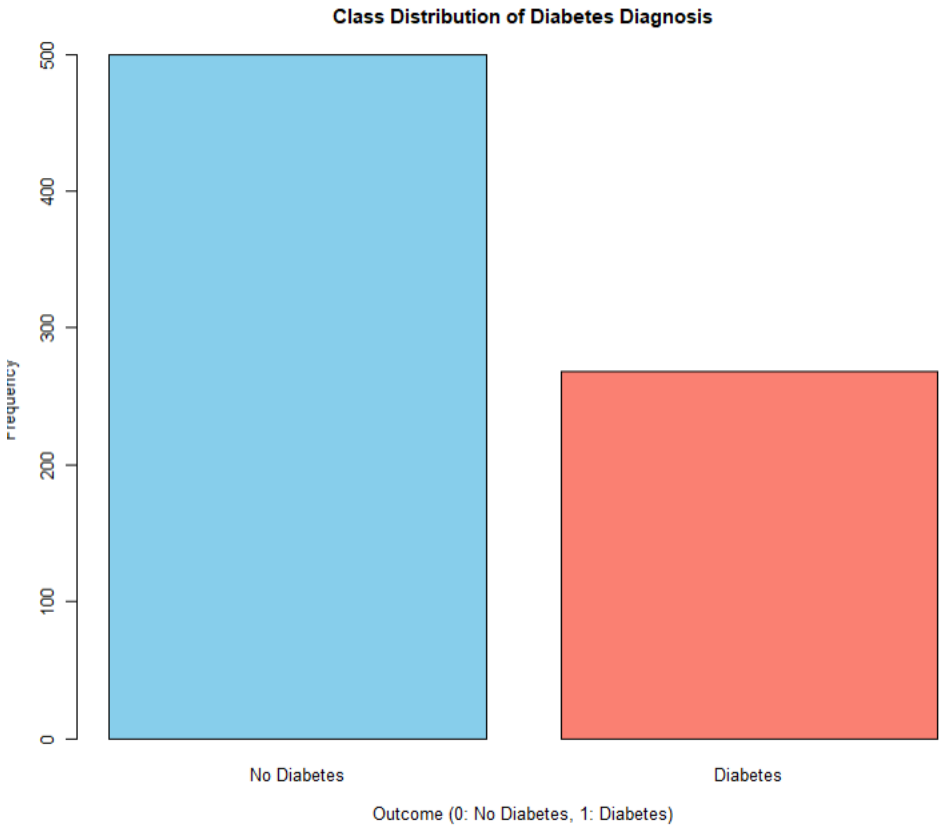




Data Exploration

1. Class distribution

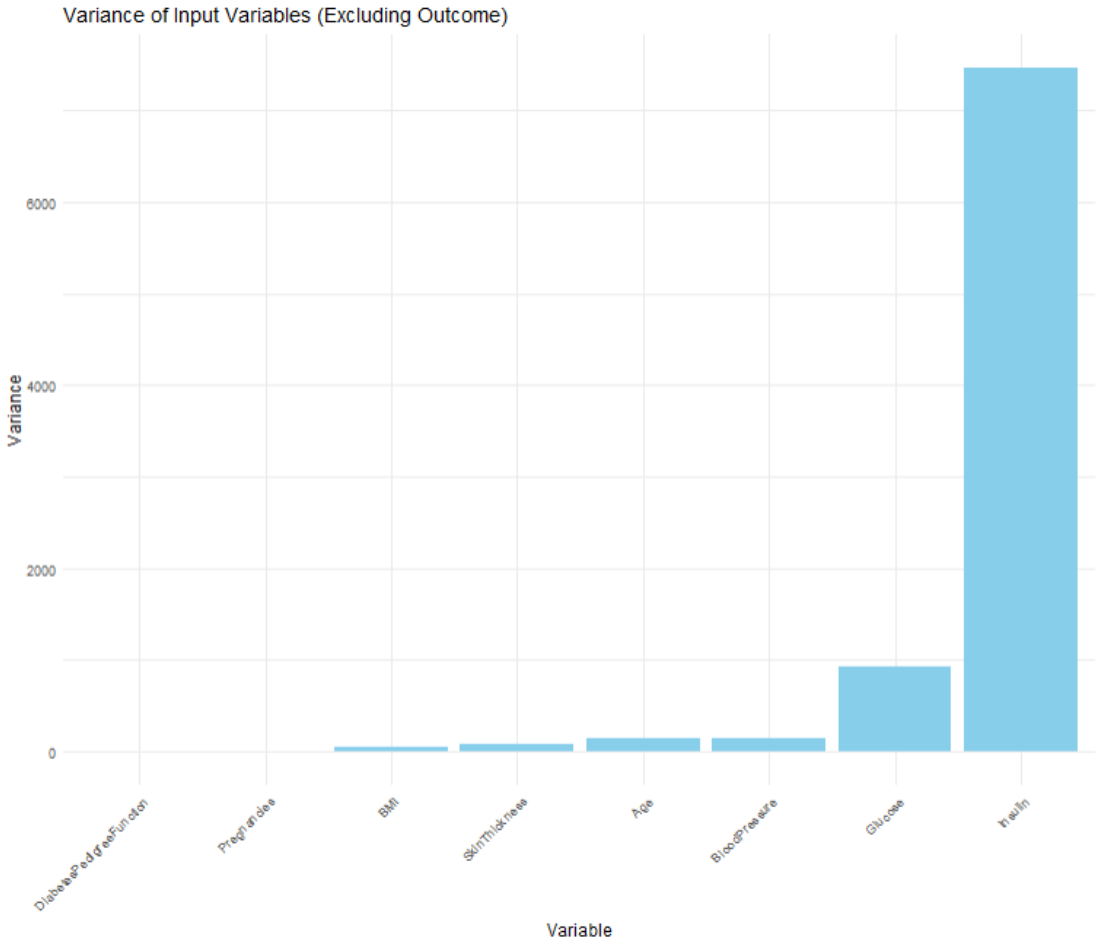
   Bar plot visualization about class distribution:



   This is not a balanced classification problem, because in a balanced scenario, the distribution would be 50%-50%. The barplot shows, that in this dataset there is about twice as many non diabetic individuals than individuals with diabetes diagnosis.

2. Variance assesment

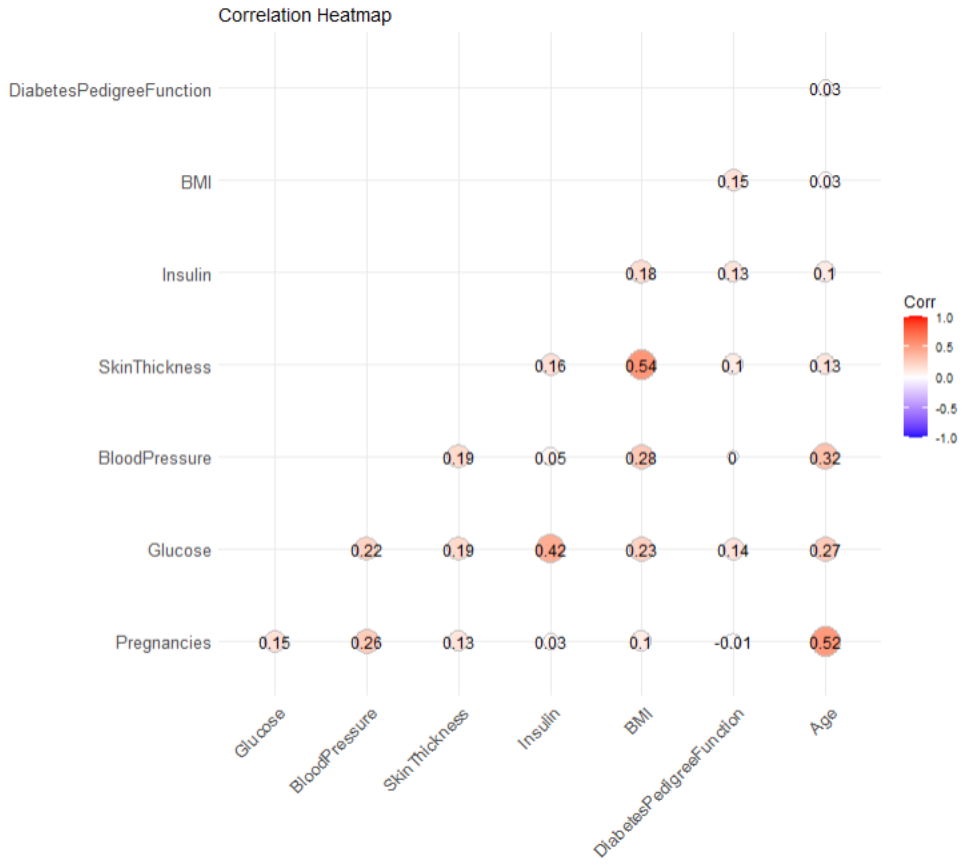
First, I visualized the variance across the variables:



The plot shows, that many variables show low variance. Especially *Pregnancies* and *DiabetesPedigreeFunction* show negligble variance, The Insulin variable appears to have the highest variance by far, followed by Glucose.
Although many variables showed low variance, I did not remove any of them, as I did not find that necessary based on the analysis results.

3. Correlation Analysis

Correlation analysis among the input variables with correlation heatmap:



The strongest positive correlation is between *SkinThickness* and *BMI*. This suggests that individuals with higher *BMI* tend to have thicker skinfold measurements, which is expected since higher *BMI* often corresponds to increased body fat.

The second strongest positive correlation is between *Age* and *Pregnancies* (0.5. This is likely because older individuals are likely to have more children.

Glucose and *Insulin* show moderate positive correlation, which tells that as glucose levels increase, insulin levels tend to decrease.

The heatmap shows that certain input variables like *Age*, *Glucose* and *BMI* might play a stronger role in the datasets analysis, as they show strongest correlation accross all variables.

Most other variables (e.g., DiabetesPedigreeFunction, SkinThickness, etc.) have weak or very weak correlations, with values ranging from 0.01 to 0.22.

Cluster Analysis

1. How K-Means Works

K-Means Algorithm Overview:

K-means is a clustering algorithm used to partition data into *k* groups or clusters. K-means clustering is designed for numerical data, because it relies on distance metrics. It works by:

1. Initalization: Randomly selectiong k intial clusters.
2. Assignment: Assigning each data point to the nearest centroid.
3. Update: Recalculating centroids as the mean of assigned points
4. Iteration: Repeating assignment and update steps until centroids stabilize or a set number of iterations is reached.
5. The result is *k* clusters with their corresponding centroids and datapoints.

Data Transformation for K-Means

Normalization

- K-Means relies on distance measures, which are sensitive to the scale of features. Features with larger ranges dominate the clustering, and normalization ensures that all features contribute equally to the clustering process. Normalization can be done by scaling the data so that each feature has a mean of 0 and a standard deviation of 1 (standardization), or scales to a range.

Handling Outliers

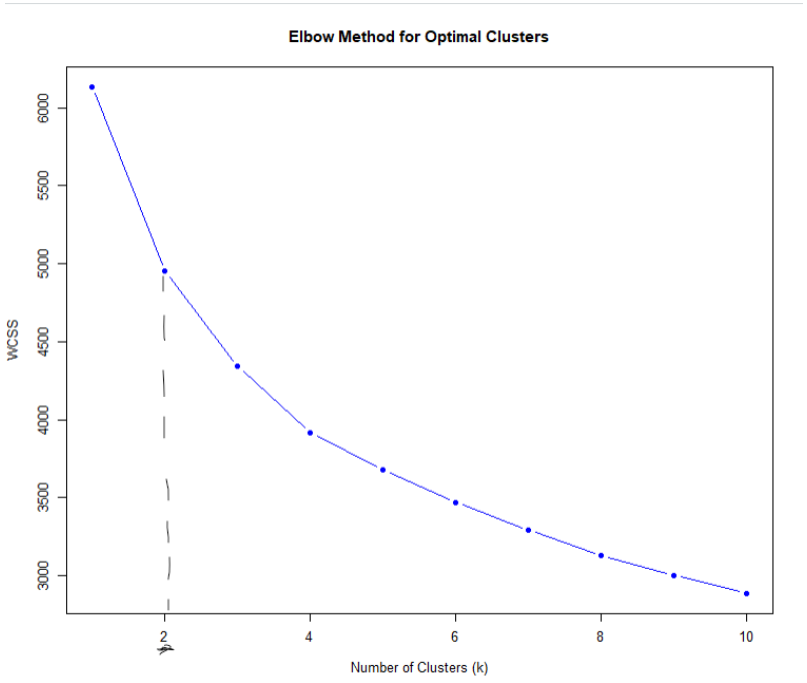
- Outliers can skew the centroids and lead to poor cluster assignments. Outliers can be handled by data transformation, removal or replacing the outliers with median or average. This creates more stable and meaningful clusters.

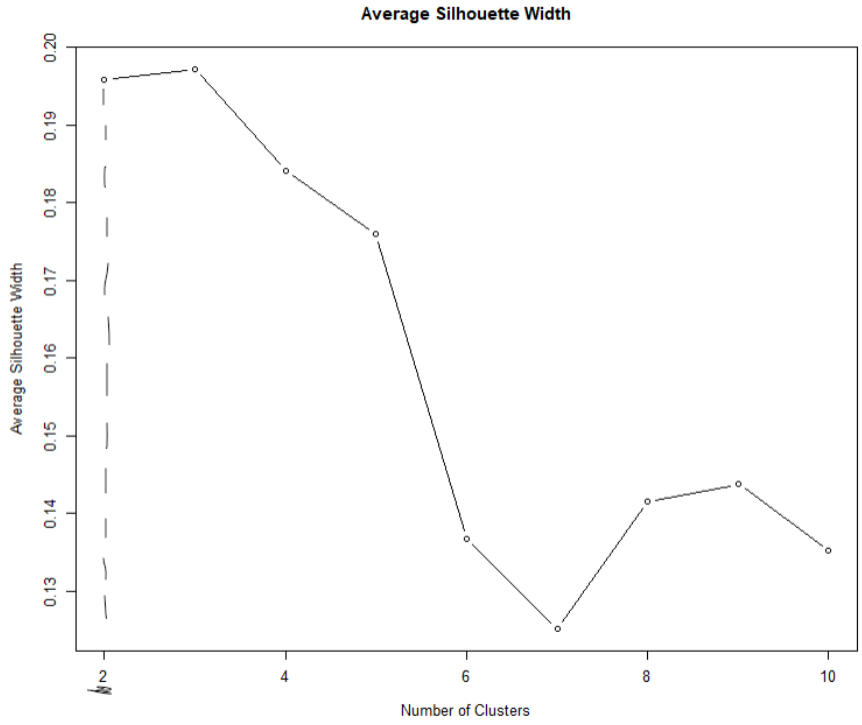
Encoding Categorical Values

- K-Means requires numerical data. Categorical values need to be converted into a suitable format, or be left out from the analysis.

2. <u>Determine the Number of Clusters</u>

   Elbow and silhouette analysis:

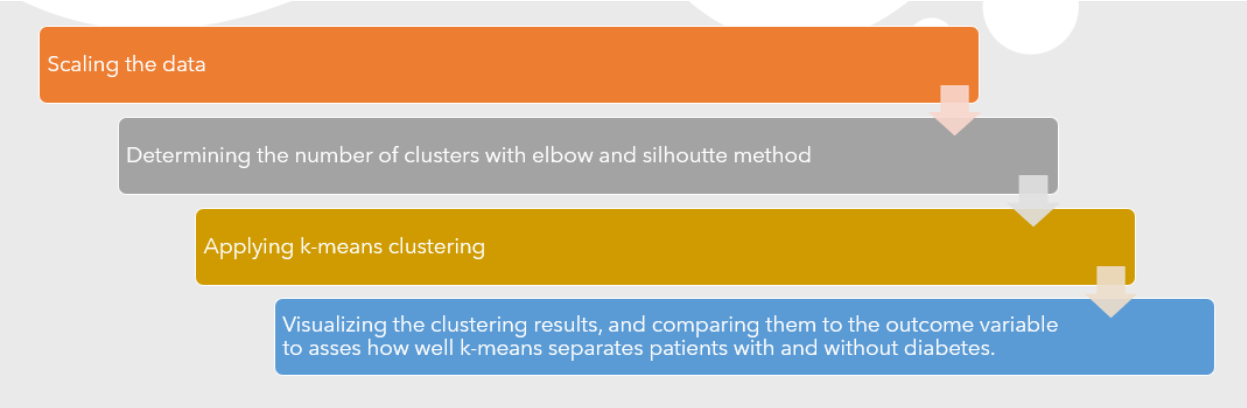




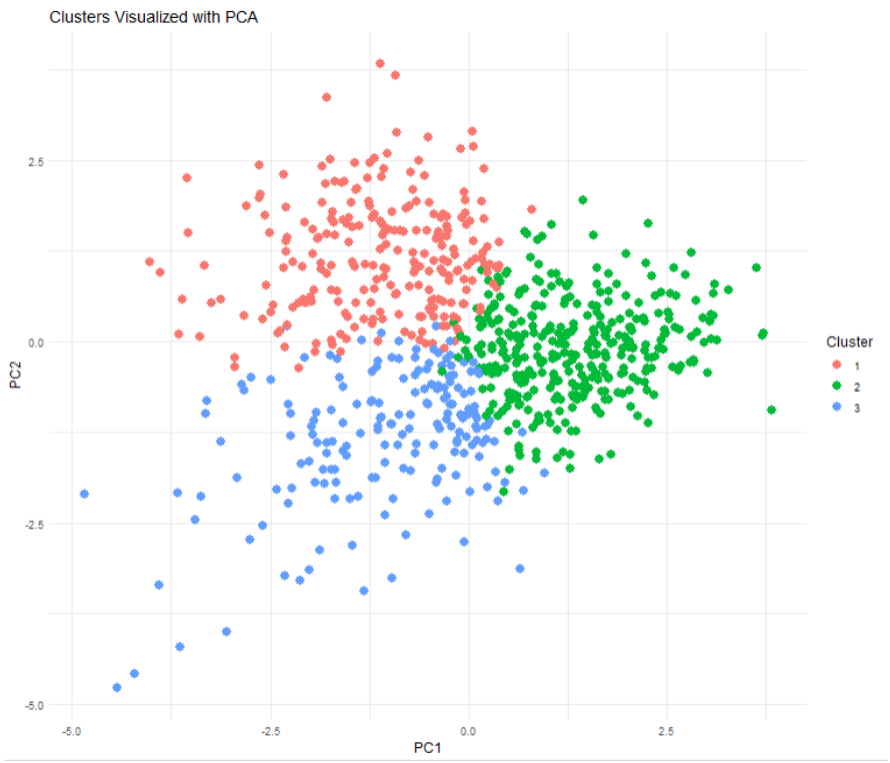
Silhouette and elbow methods suggest that the optimal k for this case would be 2. Later I ended up choosing K=3, because it was the larges number that showed clear clustering results. K=2 worked well too.

3. Apply Clustering

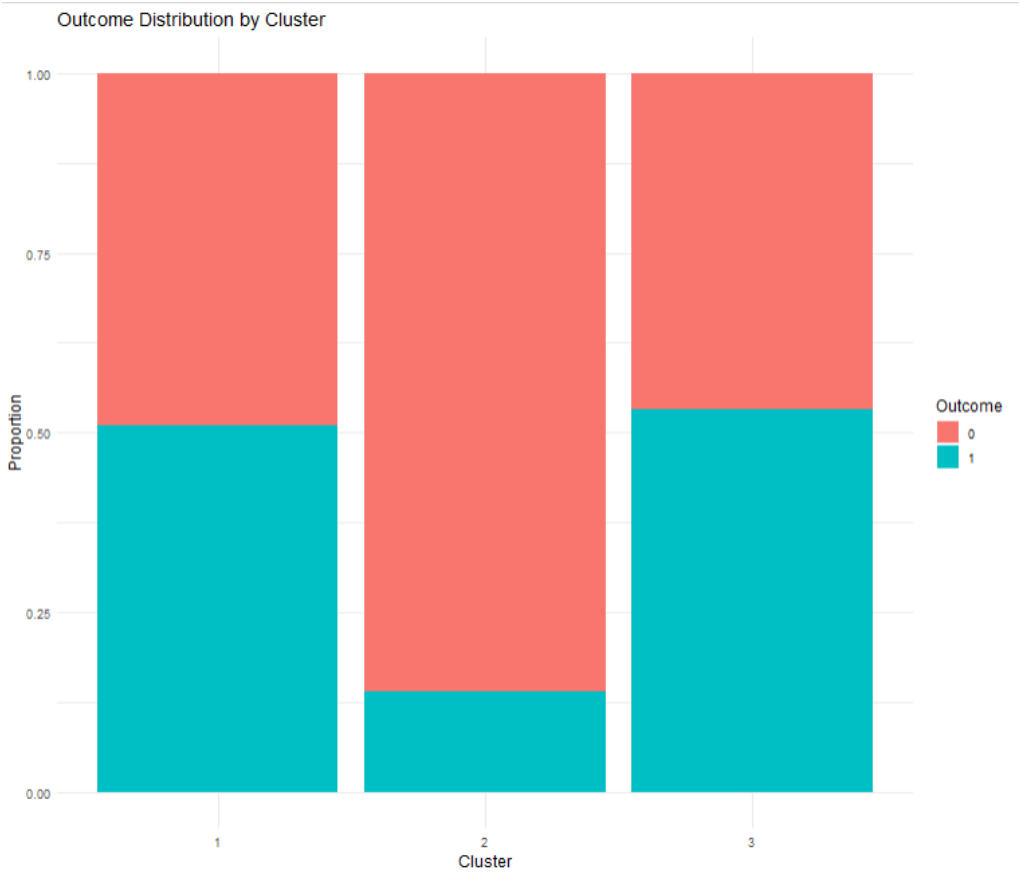
My clustering tasks:



K-Means Clustering results



PCA visualization shows, that I ended up grouping the data into 3 clusters. In PCA space, the 3 clusters datapoints are grouped clearly, without overlapping each other. This suggests, that the clustering succesfully grouped the data into three distinct clusters. The lack of overlap demonstrates the effectiveness of the clustering approach.



Outcome distripution is mostly same in clusters 1 and 3, but cluster 2 shows larger distripution of non-diabetic individuals. This tells, that clusters 1 and 3 have some other variable(s) than output Outcome that makes them different.

```
> # Contingency table comparing cluster assignments to the outcome variable to
> # show how many patients in each cluster have diabetes or not
> contingency_table <- table(diabetes_with_clusters$Cluster, diabetes_with_clusters$Outcome)
> print(contingency_table)

    0   1
1 118 123
2 297  48
3  85  97
>
```

Contigeny table also shows the outcome distribution of each cluster by showing the exact amounts. The distribution suggests, that the k=2 cluster would have seperated the clusters by Outcome better. However, cluster 1 and 3 have other variables, that make them different form each other.

```
> # Profile the clusters by comparing means of input variables
> cluster_profile <- aggregate(diabetes_with_clusters[, c("Pregnancies", "Glucose", "BloodPressure", "SkinThickness", "Insulin", "BMI", "Diabe
tesPedigreeFunction", "Age")],
+                              by = list(cluster = diabetes_with_clusters$Cluster),
+                              FUN = mean)
> print(cluster_profile)
  cluster Pregnancies  Glucose BloodPressure SkinThickness  Insulin      BMI DiabetesPedigreeFunction      Age
1       1    7.489627 130.7552      78.35270      29.81328 138.5228 32.55643                0.4639917 46.41909
2       2    2.881159 107.0493      67.04058      24.26377 114.7449 28.67768                0.4181768 25.93623
3       3    3.285714 137.2967      74.62088      37.35714 192.6648 39.48187                0.5841099 29.63736
>
```

Table to show means of input variables in each cluster. This was a nice way to see the difference between the clusters. While cluster 1 and 2 did not differ by outcome distribution, but did differ significantly by the amount of pregnancies, insulin and age. Cluster 3 also showed largest mean of BMI, SkinThickness, Insulin and DiabetesPedigreeFunction values. Cluster 1 has significantly larger mean values of age and number of pregnancies, when compared to other clusters. The values were ≈7,5 pregnancies and age of 46,4. Also, in cluster 1 there was highest mean BloodPressure.

Classification Analysis: KNN to predict diabetes

K-Nearest Neighbours (KNN) algorithm is a classification method. It works as follows:

- Training phase: KNN stores the entire training dataset
- Prediction Phase: Classifying a new datapoint
  1. Calculatong the distence between the new datapoint and all points in the training set.
  2. Identifyin the k nearest neighbors to the data point
  3. Assigning the class label based on the majority class among the k neighbors.
- KNN assumes thet data points close to each others are likely to belong to the same class

<u>Advantages and DrawBacks of KNN Classification</u>

Advantages:

- It is simple: easy to understand and implement
- No training phase: it is quick to set up as it stores the data directly
- Can be used for both classification and regression

Drawbacks:

- Computational cost: slow prediction for large datasets, because distance has to be calculated to all datapoints

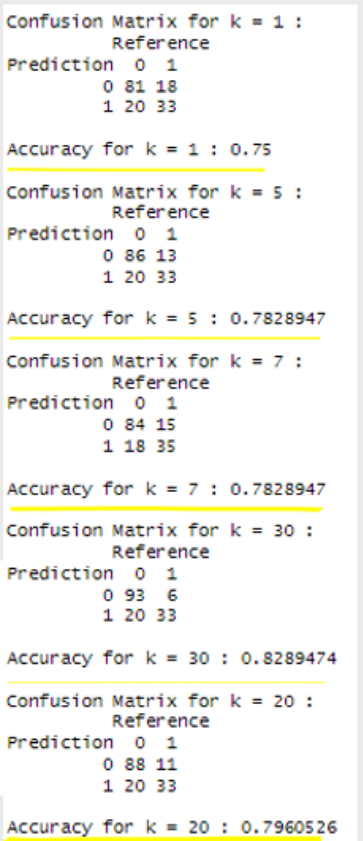
- Sensitive to Irrelevant features, preformance may degrade if irrelevant or redundant values are present
- High-dimencional data can make distances less meaningful, which can reduce accuracy.
- KNN may struggle with imbalanced datasets

A confusion matrix is a table that summarizes the performance by classification model by comparing predicted and actual outcomes. By analyzing the confusion matrix, you can evaluate the KNN model's effectiveness in predicting diabetes.

My KNN Classification:

1. I first splitted the dataset into training(80%) and testing (20%) sets.
2. Training KNN classifiers usinf five different values of k.

Confusion matrixes for each experiment:



```
Confusion Matrix for k = 1 :
          Reference
Prediction  0  1
         0 81 18
         1 20 33

Accuracy for k = 1 : 0.75

Confusion Matrix for k = 5 :
          Reference
Prediction  0  1
         0 86 13
         1 20 33

Accuracy for k = 5 : 0.7828947

Confusion Matrix for k = 7 :
          Reference
Prediction  0  1
         0 84 15
         1 18 35

Accuracy for k = 7 : 0.7828947

Confusion Matrix for k = 30 :
          Reference
Prediction  0  1
         0 93  6
         1 20 33

Accuracy for k = 30 : 0.8289474

Confusion Matrix for k = 20 :
          Reference
Prediction  0  1
         0 88 11
         1 20 33

Accuracy for k = 20 : 0.7960526
```

K=30 showed the best accuracy (≈0.823.) The performance seems to improve, and predictions come more robust and stable as the value of k increase, this can be seen in the accuracies for each k.

Discussion: Clusteriong vs. Classification

By combining clustering and classification, richer insights can be obtained: clustering can reveal subgroups for further classification analysis, enhancing prediction models and understanding complex datasets.

Outcomes of Clustering (K-means)

- **Purpose**: Clustering algorithms like K-means group data into distinct clusters. This helps in identifying natural patterns or structures in the data.
- **Insights**:
    - Grouped the dataset into 3 clusters based on similar input features.
    - Cluster profiling revealed key differences in variable means, e.g., variations in glucose levels or BMI across clusters.
    - Outcome distribution within clusters (e.g., diabetes vs no diabetes) provided insights into the likelihood of diabetes for each group.
    - PCA visualization highlighted separability of clusters in lower dimensions.
- **Utility**
    - Useful for exploratory data analysis and discovering hidden patterns.
    - Can guide targeted interventions, such as focusing on at-risk groups identified by clustering.

Outcomes of Classification (KNN)

- **Purpose**: Classification algorithms like KNN predict a target variable (e.g., diabetes) based on labeled data.
- **Insights**:
    - The confusion matrix evaluated the model's performance by showing true positives, false positives, true negatives, and false negatives.
    - Accuracy for different values of *k* highlighted the sensitivity of KNN to parameter tuning.
    - Higher values of *k* resulted in more stable predictions

- **Utility**:
    - Effective for predicting specific outcomes, such as whether a patient has diabetes.
    - Directly applicable to decision-making tasks, such as diagnosing conditions based on input features.