Post Graduate Program - Data Science

In Partnership With Purdue University

Course Project – Data Science Capstone

Healthcare PGP

Submitted by:

Lavkush Singh

Submitted to:

Purdue University – Simplilearn

Agenda

- Introduction
- Dataset Summary
- Exploratory Data Analysis
- Missing Values Analysis
- Correlation Analysis
- Outlier Analysis
- Data Scaling and Principle Component Analysis
- Predictive Modelling Analysis
- Predictive Modelling Analysis: KNN Classifier
- Predictive Modelling Analysis: Random Forest Classifier
- Performance Metrics Analysis
- Tableau Report
- Appendix

Introduction

- NIDDK (National Institute of Diabetes and Digestive and Kidney Diseases) research creates knowledge about and treatments for the most chronic, costly, and consequential diseases. The dataset used in this project is originally from NIDDK.
- The objective is to predict whether or not a patient has diabetes, based on certain diagnostic measurements included in the dataset.
- The datasets consists of several medical predictor variables and one target variable (Outcome). Predictor variables includes the number of pregnancies the patient has had, their BMI, insulin level, age, and more.
- This is **Supervised – Classification** Problem, with 2 levels of outcome to predict.

Dataset Summary

- Single file having the dataset, with the name – “health care diabetes.xlsx”
- There are 8 features (columns) and 1 output column to predict.
- Data-type of variables are “int” (6 columns excluding target) and “float” (2 columns).
- “output” (target) column is of “int” type, with 2 level of output (0 --> non-diabetic, 1 --> diabetic).

Exploratory Data Analysis

Understanding Data – Datatypes, Dimension, Null Values Summary

```
diabetes_data.info()  # understanding column wise datatype and null values
```

```
<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)
```

Understanding Data – Datatypes, Dimension, Null Values Summary

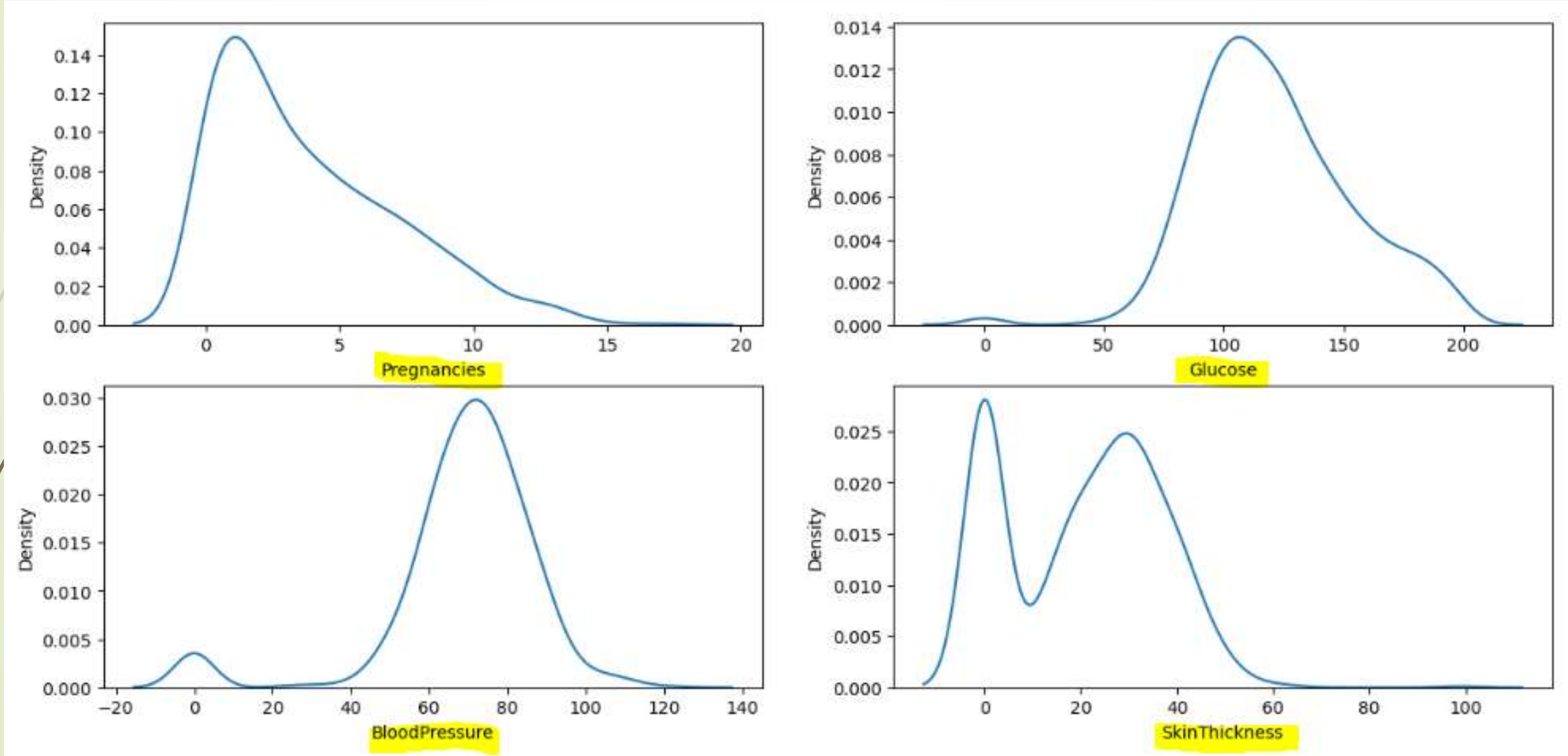
Missing Value – Count per column

Glucose	5
BloodPressure	35
SkinThickness	227
Insulin	374
BMI	11
dtype: int64	

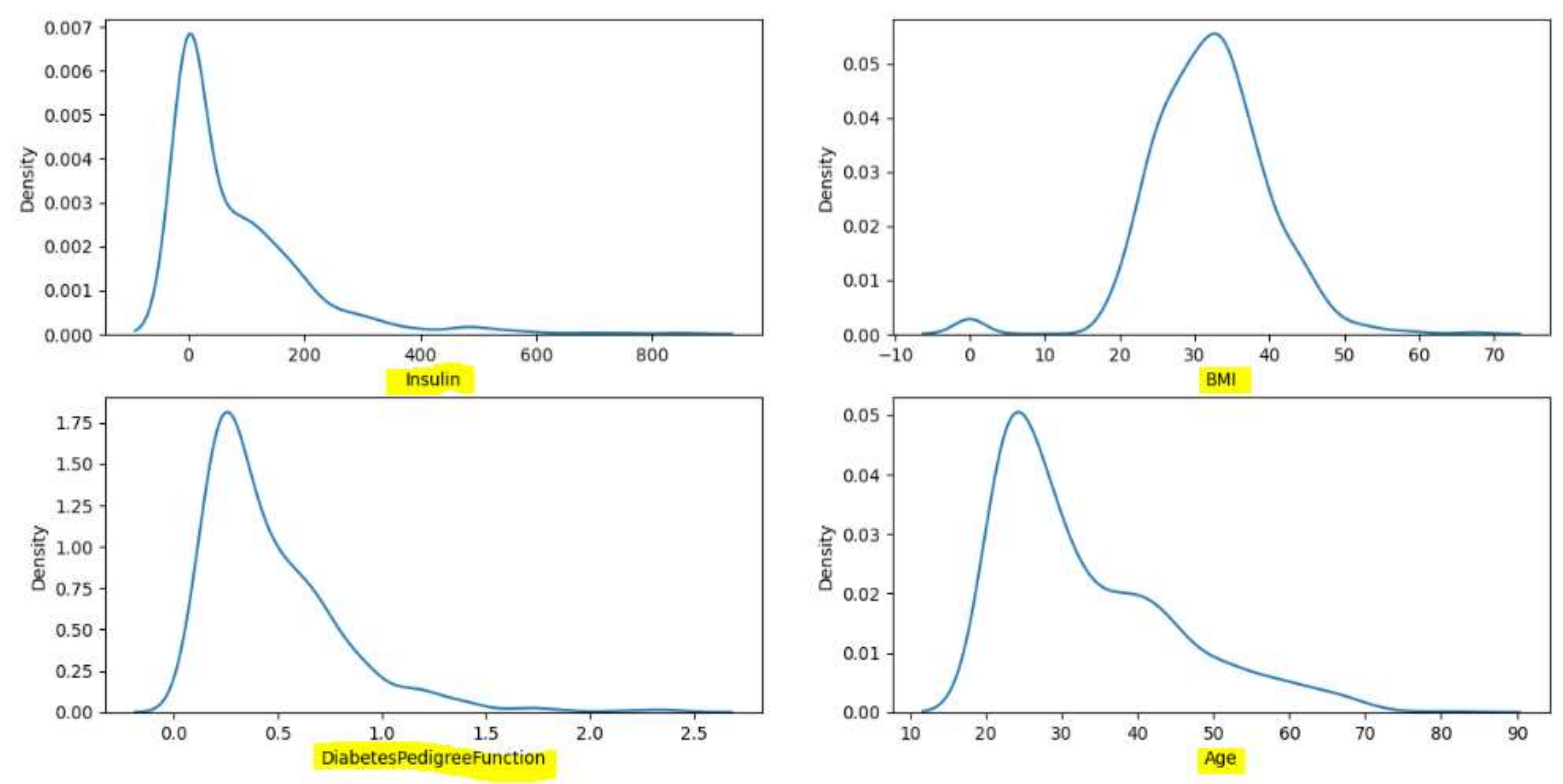
Missing Values – Percent (%) per column

Glucose	0.65
BloodPressure	4.56
SkinThickness	29.56
Insulin	48.70
BMI	1.43
dtype: float64	

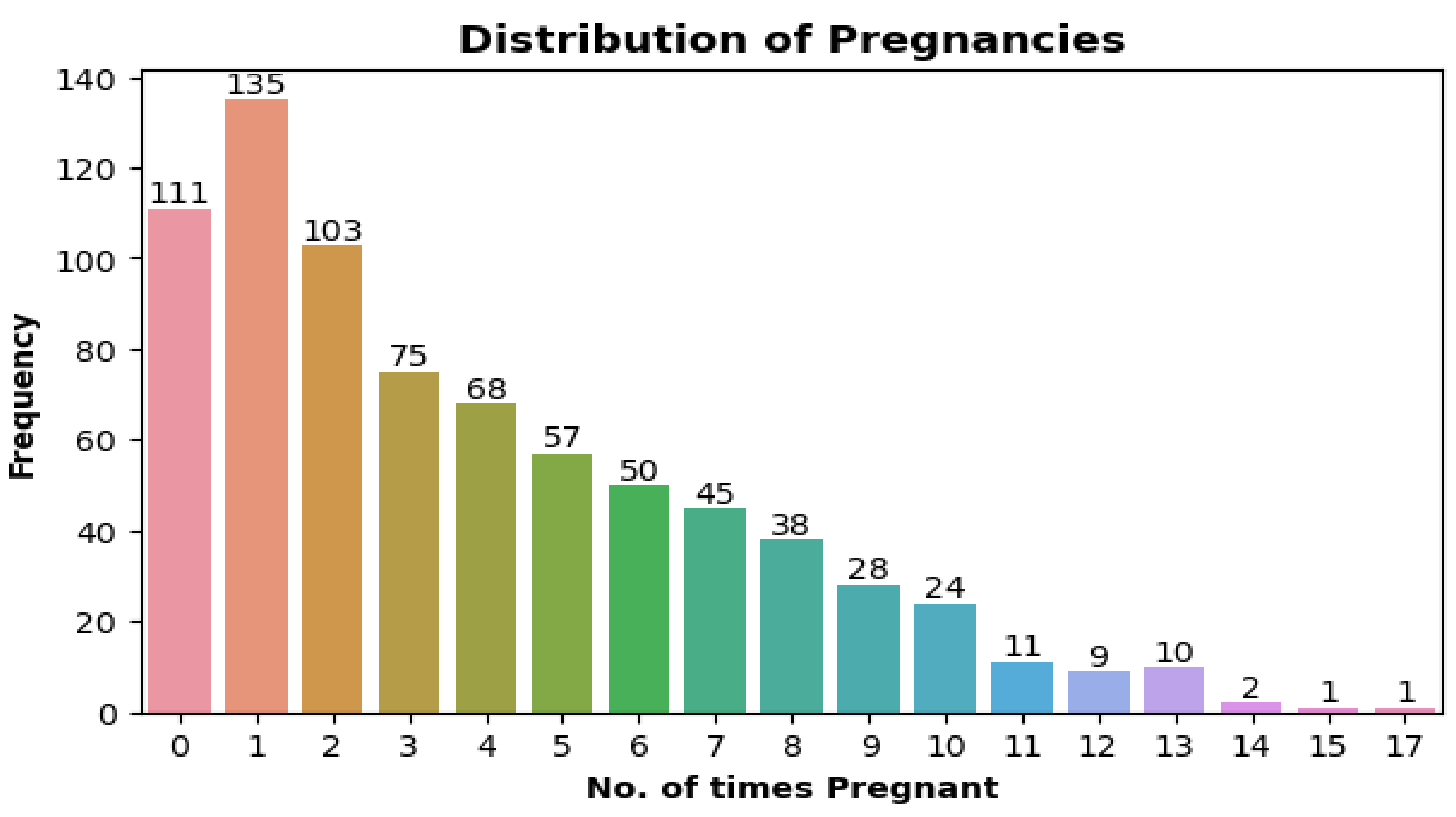
KDE Distribution



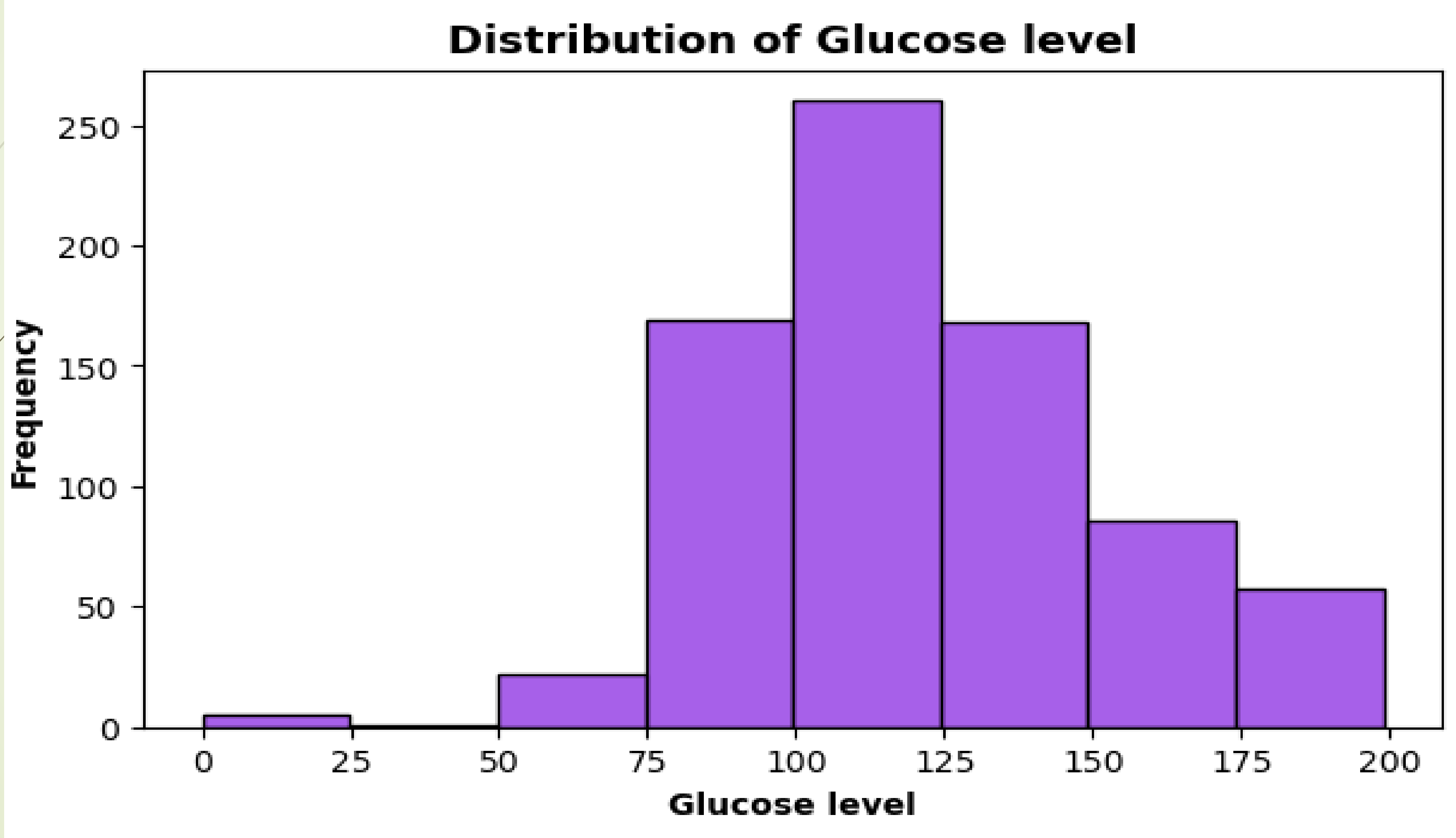
KDE Distribution



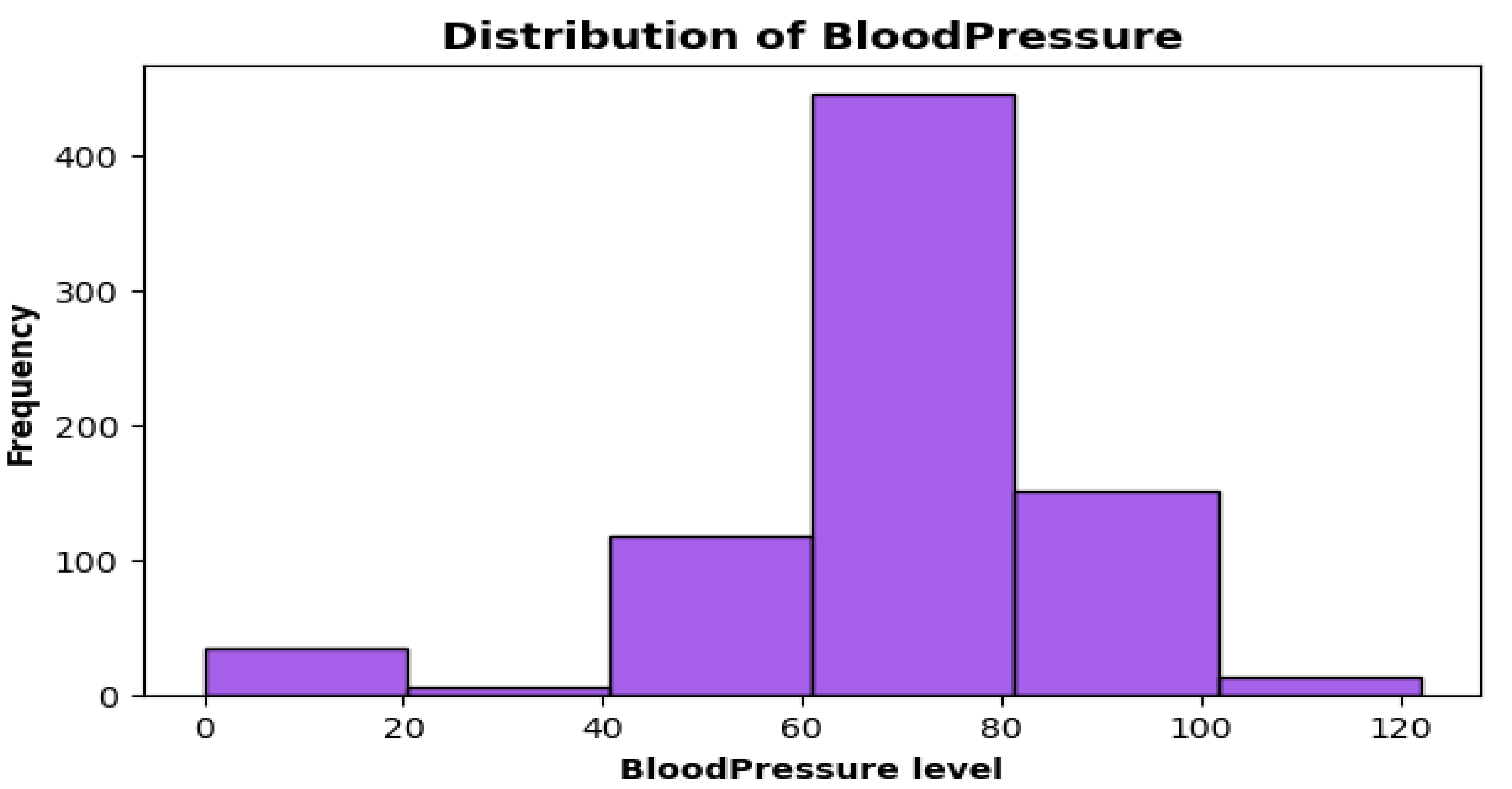
Distribution – 'Pregnancies'



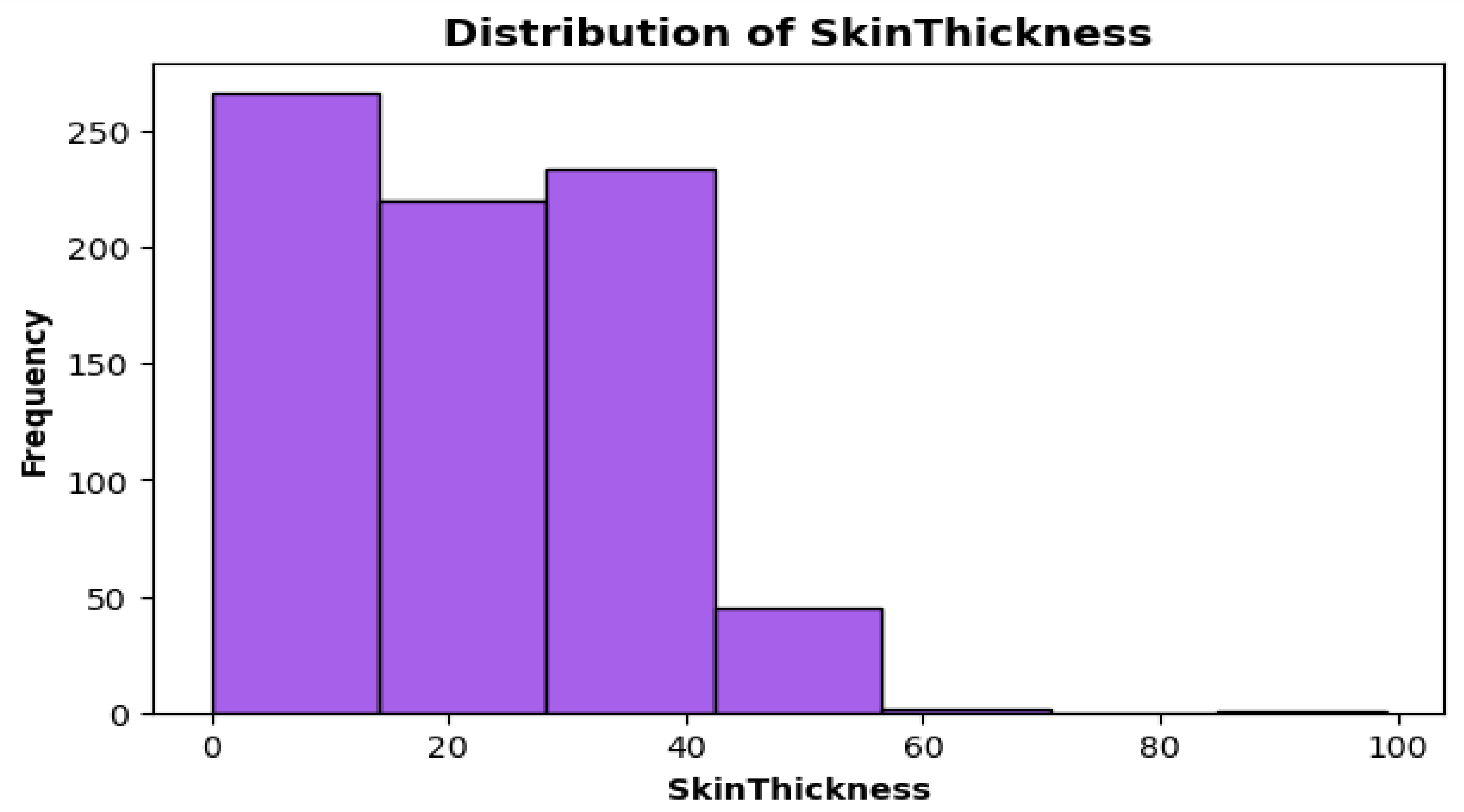
Distribution – ‘Glucose’



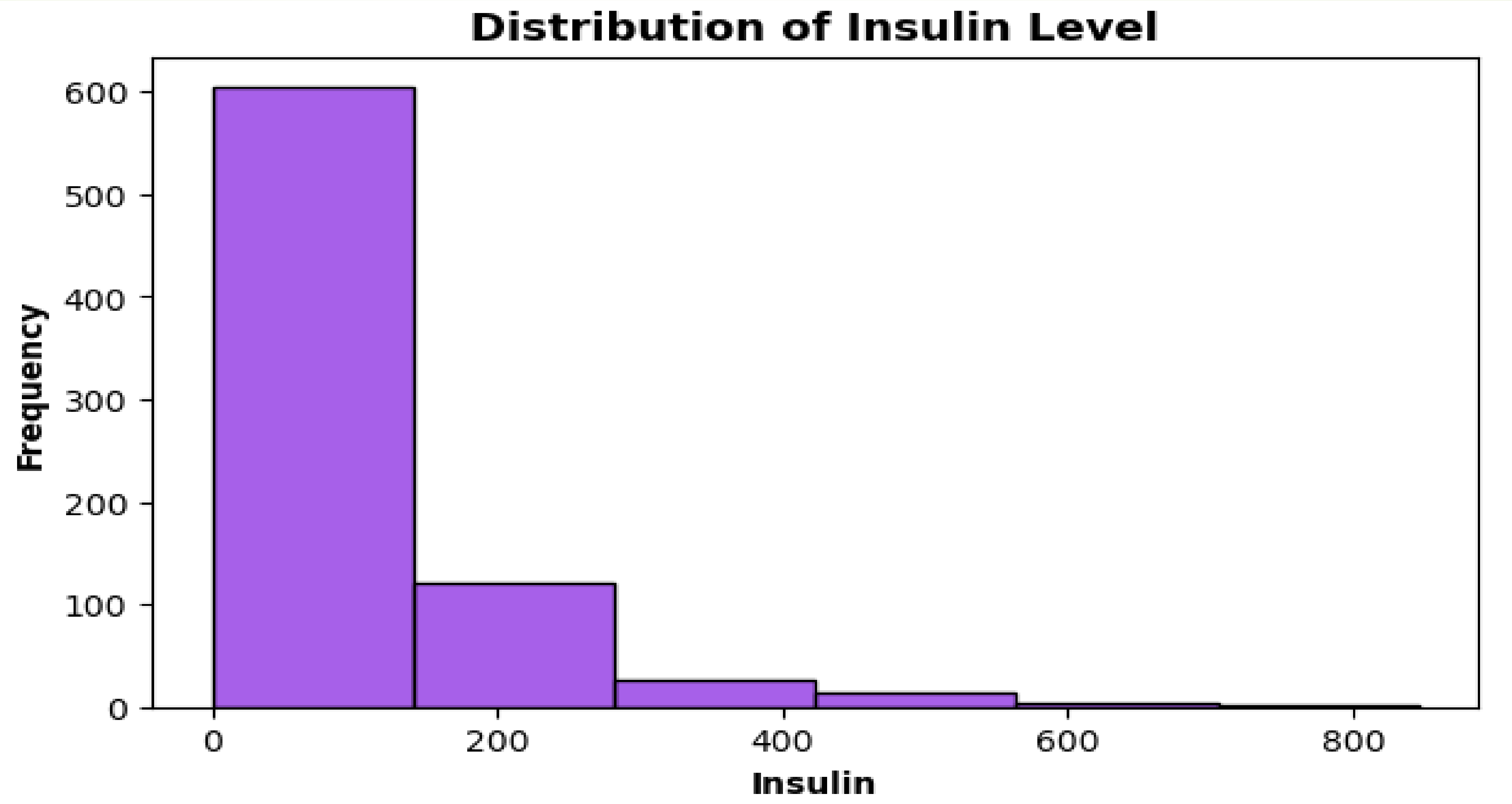
Distribution – ‘BloodPressure’



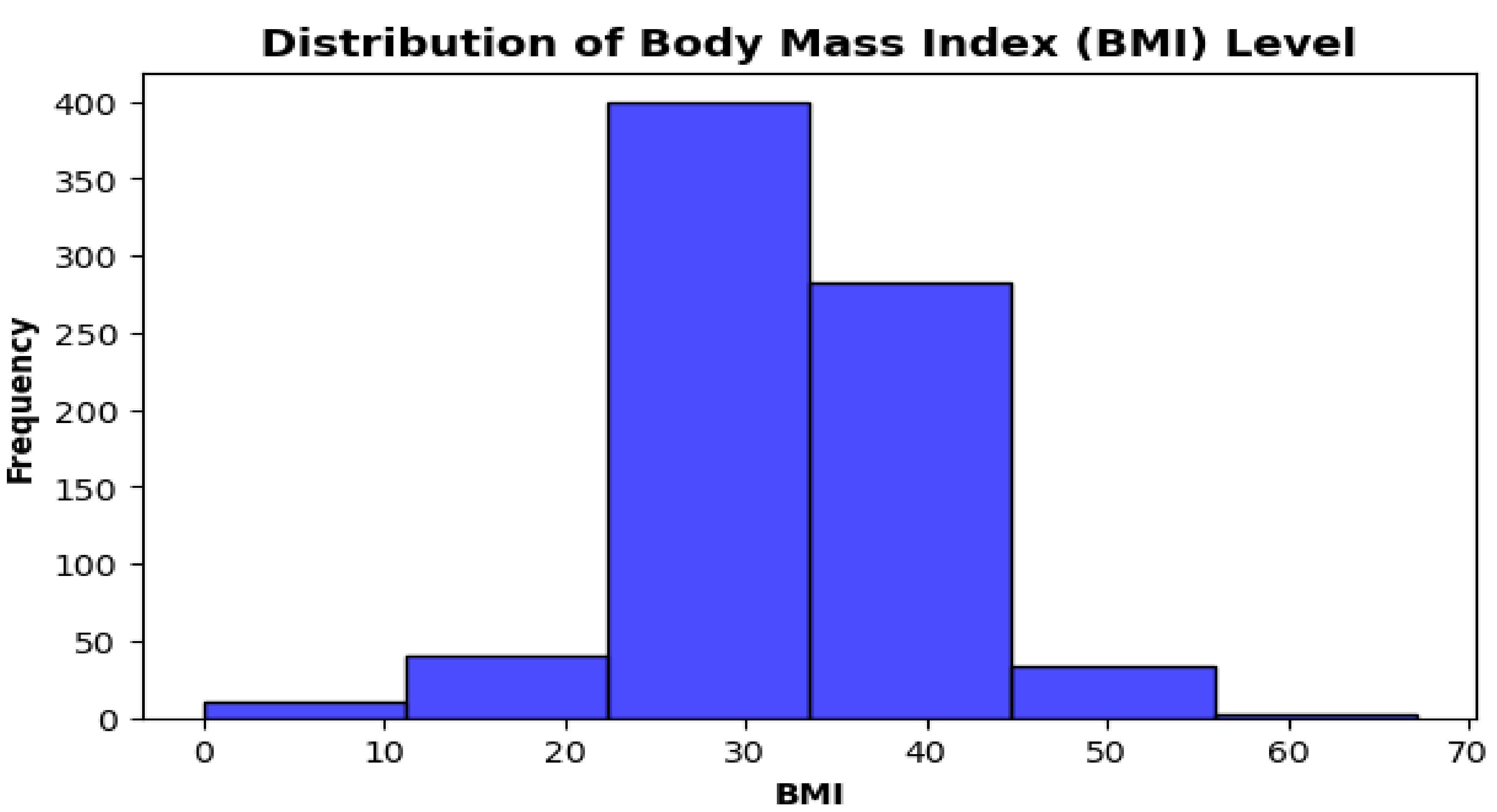
Distribution – 'SkinThickness'



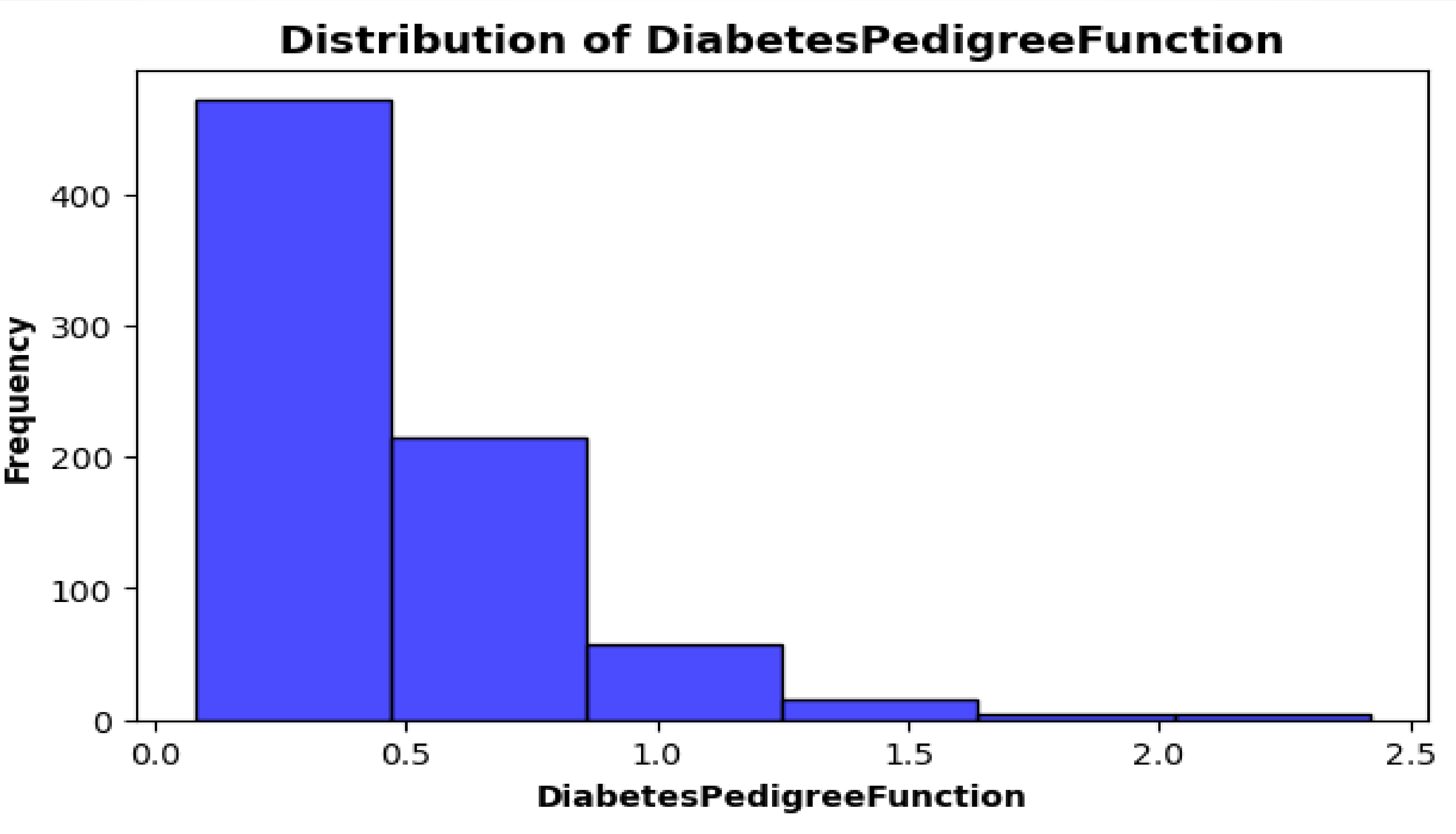
Distribution – 'Insulin'



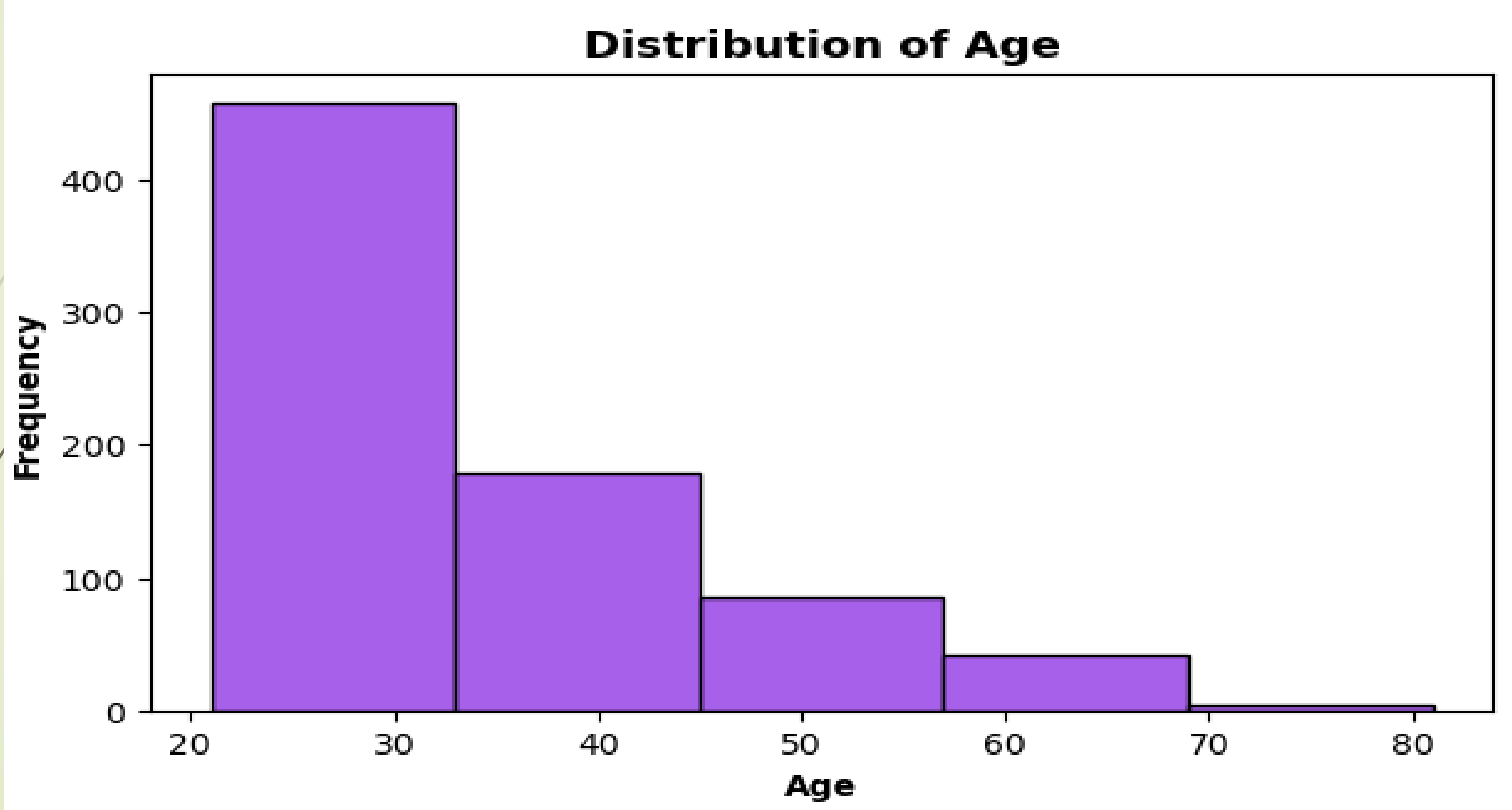
Distribution – 'BMI'



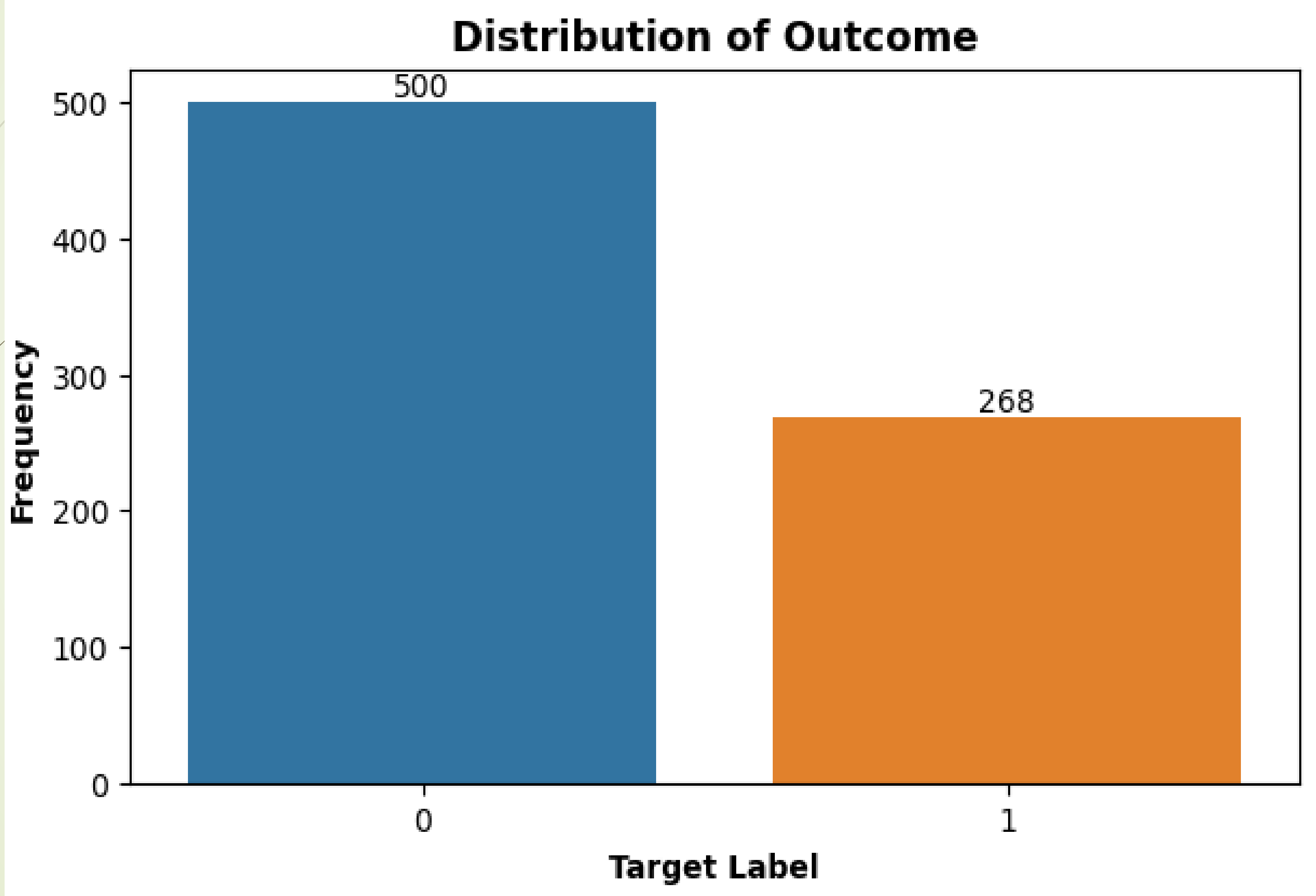
Distribution – 'DiabetesPedigreeFunction'



Distribution – ‘Age’



Distribution – ‘Outcome’



Exploratory Analysis Summary

- We have **9 features** (columns) and **768 observations** (rows or entries) in dataset.
- There are no null values in the dataset, however, 0 value in ['Glucose', 'BloodPressure', 'SkinThickness', 'Insulin', 'BMI'] columns have been treated as missing value.
- Most of the people are between **21** to **33 years** of age group.], where as very few are of **69 years** or older.
- 'Glucose', 'BloodPressure', 'BMI' columns has approximately **normal distribution**, while other features have **skewed distributions**
- Of the total 768 observations, **14.45%** (111 observations) comprises of the persons who have been pregnant.

Exploratory Analysis Summary

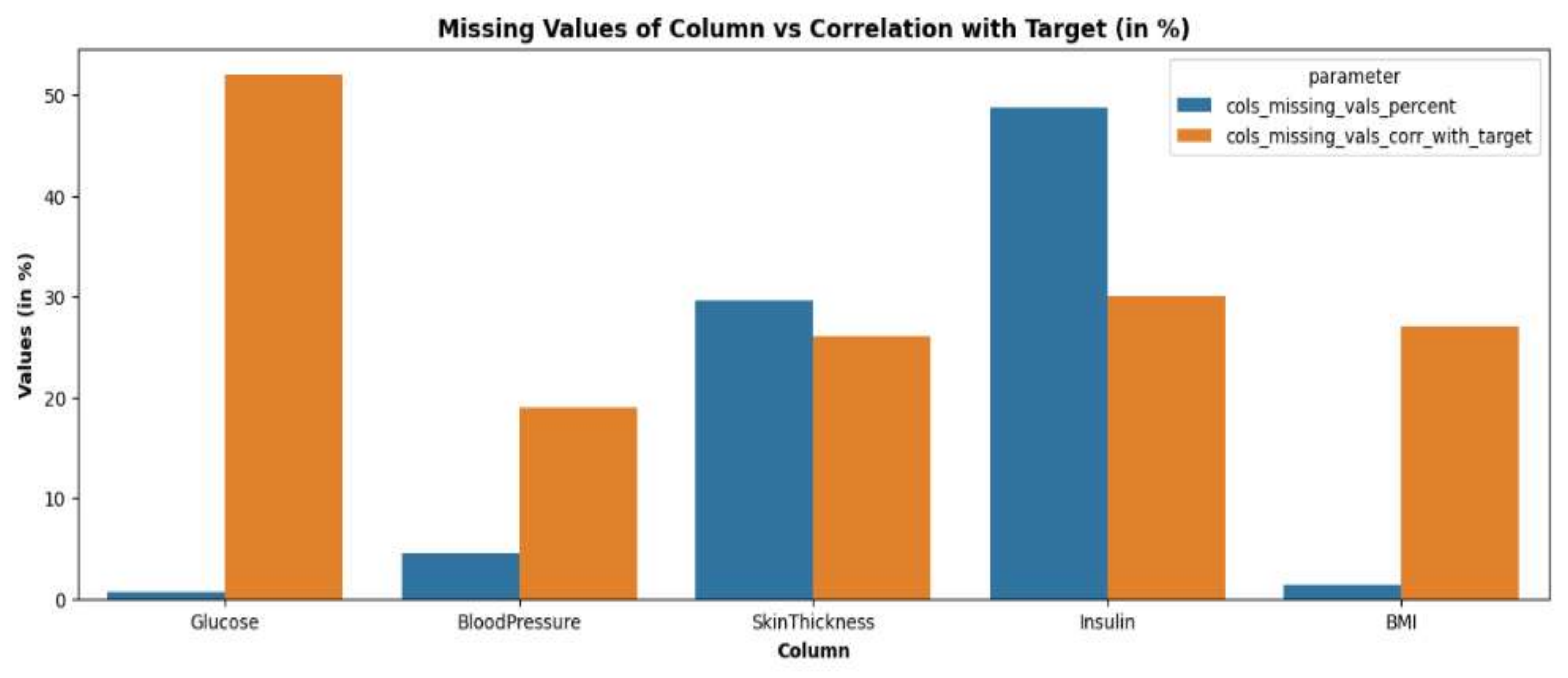
- This data may consists of males, however this cannot be verified since the data given does not have ‘Gender’ column to determine. But the **85.55%** (657 observations) of the data is for women, because they have been pregnant for at least once.
- Among Pregnancies, Maximum Number of people has been pregnant for at least once.
- Second highest number of people are those who have never been pregnant, this might also include the number of males in the consideration; however 'Gender' of the dataset is not given.

Missing Values Analysis

Missing values and Correlation with Target (in %)

	column	cols_missing_vals_percent	cols_missing_vals_corr_with_target
0	Glucose	0.65	52.0
1	BloodPressure	4.56	19.0
2	SkinThickness	29.56	26.0
3	Insulin	48.70	30.0
4	BMI	1.43	27.0

Missing values and Correlation with Target (in %)



Missing Values Treatment

- As per the given information, ['Glucose', 'BloodPressure', 'SkinThickness', 'Insulin', 'BMI’] columns with **0** as datapoint value is to be treated as missing value
- Columns ‘SkinThickness’,’ Insulin’ has maximum of the **missing values**, of **29.56%** and **48.70%** respectively.
- Columns ‘Glucose’, ‘BloodPressure’, ‘BMI’ has relatively negligible/lower missing values count, of about **0.65%**, **4.56%** and **1.43%** respectively.
- It was observed that among the columns having missing values, the columns (after removing missing values) Glucose is **52%** correlated with the target. The colums which has maximum of the missing values, ‘SkinThickness’,’ Insulin’ have **26%** and **30%** correlation with the target variable.
- Missing values of the columns **['Glucose', 'Insulin']** were imputed with **median** and **['BloodPressure', 'SkinThickness', 'BMI']** was imputed with **mean**, keeping in mind the distribution and outliers presence.

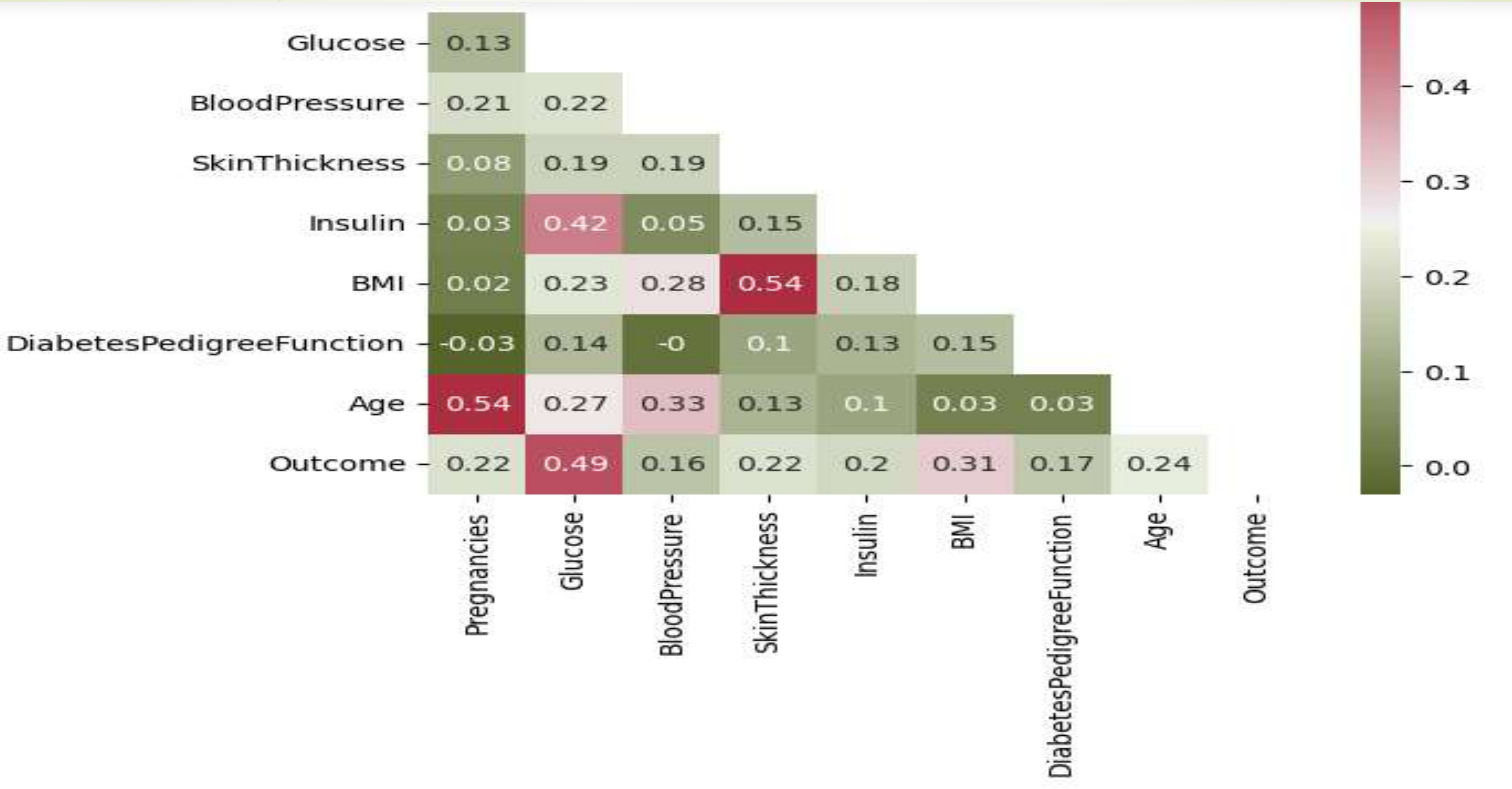
Correlation Analysis

Correlation Analysis - Matrix

```
corr_data = diabetes_data_missing_imputed.corr().round(2)
corr_data
```

	Pregnancies	Glucose	BloodPressure	SkinThickness	Insulin	BMI	DiabetesPedigreeFunction	Age	Outcome
Pregnancies	1.00	0.13	0.21	0.08	0.03	0.02	-0.03	0.54	0.22
Glucose	0.13	1.00	0.22	0.19	0.42	0.23	0.14	0.27	0.49
BloodPressure	0.21	0.22	1.00	0.19	0.05	0.28	-0.00	0.33	0.16
SkinThickness	0.08	0.19	0.19	1.00	0.15	0.54	0.10	0.13	0.22
Insulin	0.03	0.42	0.05	0.15	1.00	0.18	0.13	0.10	0.20
BMI	0.02	0.23	0.28	0.54	0.18	1.00	0.15	0.03	0.31
DiabetesPedigreeFunction	-0.03	0.14	-0.00	0.10	0.13	0.15	1.00	0.03	0.17
Age	0.54	0.27	0.33	0.13	0.10	0.03	0.03	1.00	0.24
Outcome	0.22	0.49	0.16	0.22	0.20	0.31	0.17	0.24	1.00

Correlation Analysis - Heatmap

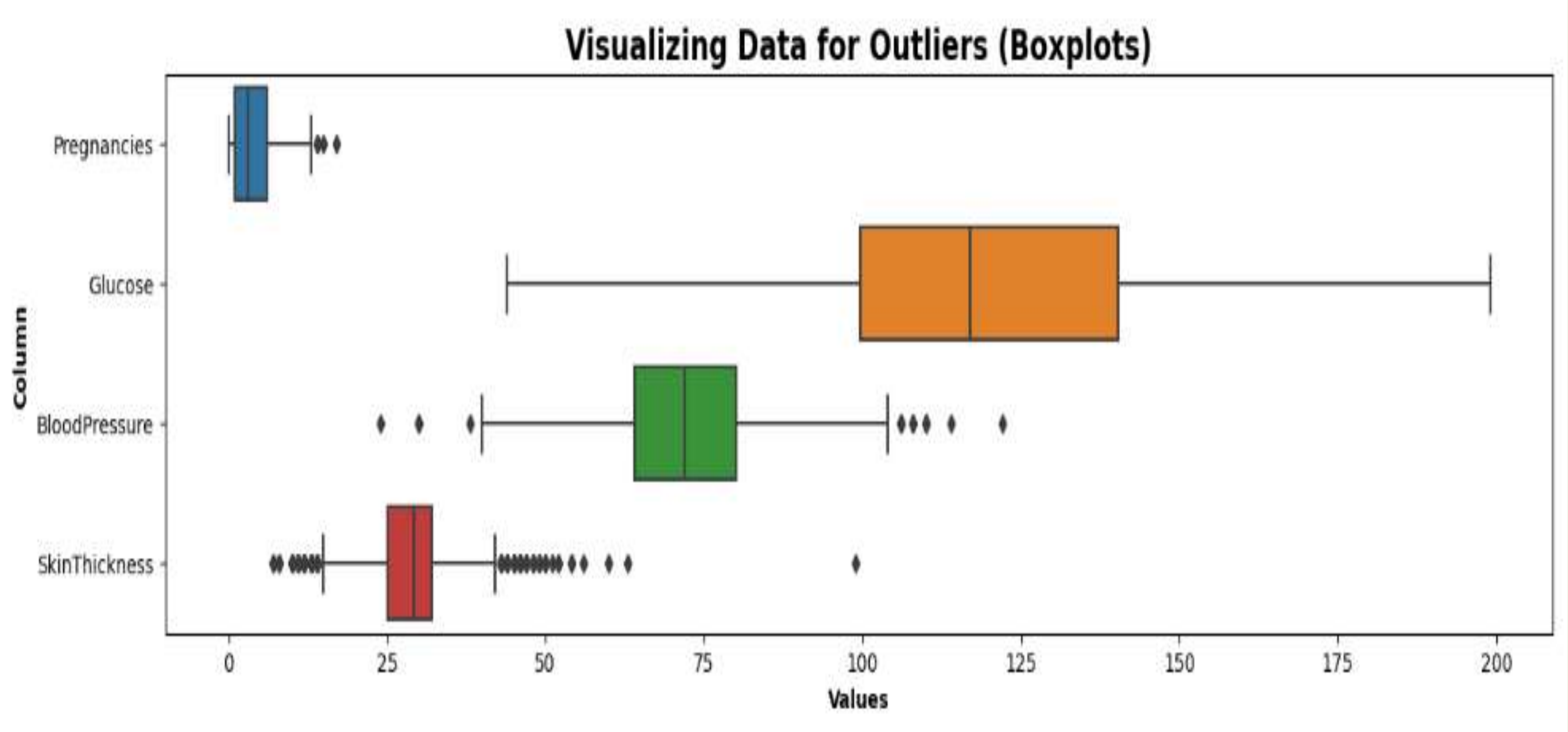


Correlation Analysis - Heatmap

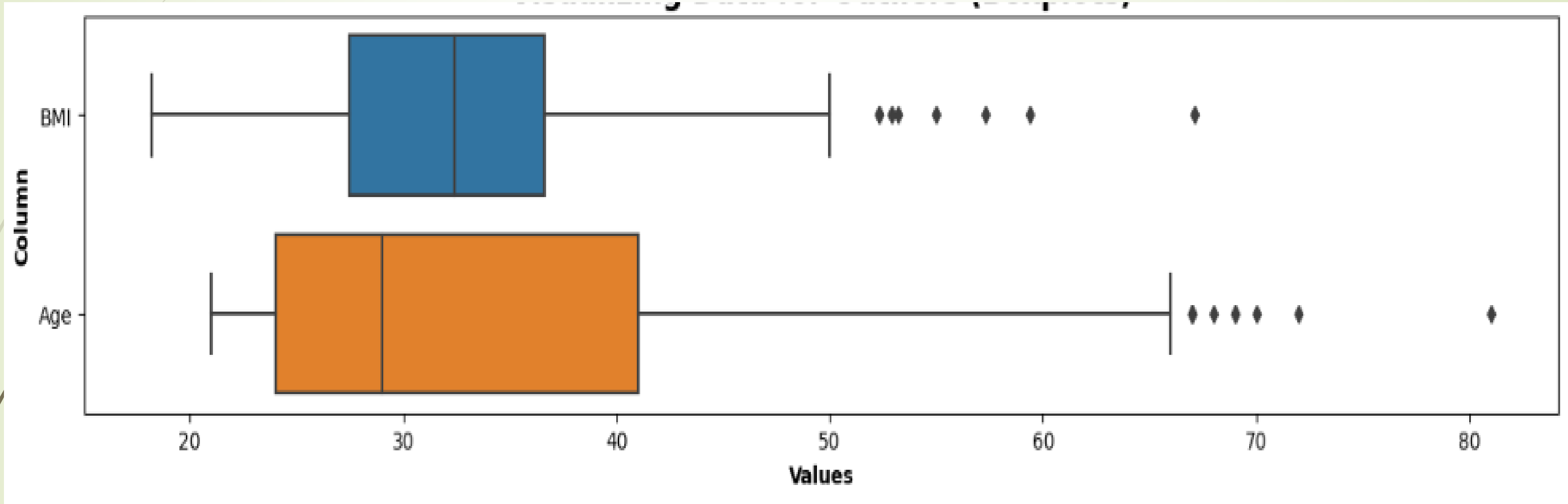
- Visualizing Correlation Matrix as heatmap, it was found that the following columns were related:
  - Glucose with Insulin
  - BMI with Skinthickness
  - Pregnancies with Age
- However, because the values were less than 0.6, the above correlations were disregarded.

Outlier Analysis

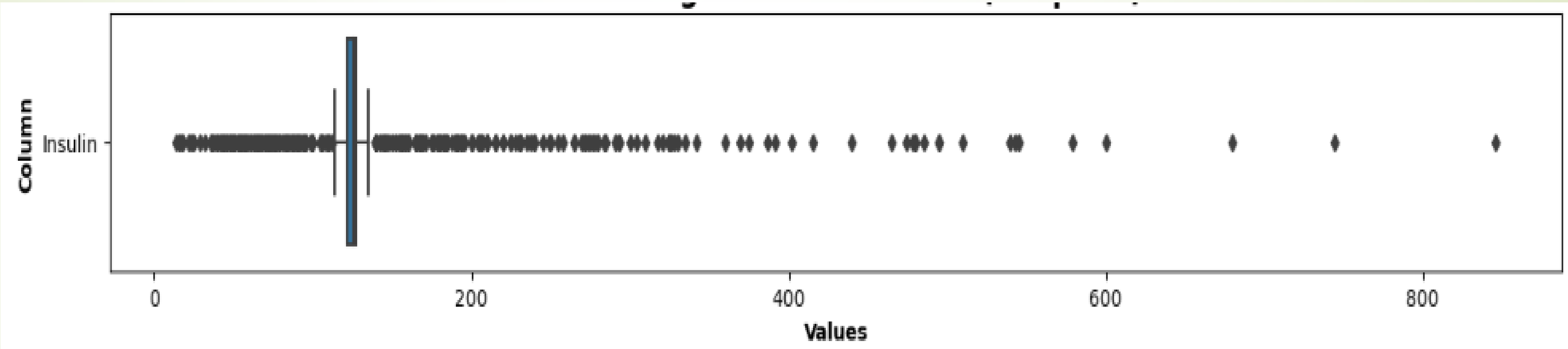
Outliers Visualization



Outliers Visualization



Outliers Visualization



Outliers Visualization

Outlier Stats (Missing Values were treated)

	no_of_outliers	no_of_outliers_percent
Pregnancies	4.0	0.52
Glucose	0.0	0.00
BloodPressure	14.0	1.82
SkinThickness	87.0	11.33
Insulin	346.0	45.05
BMI	8.0	1.04
DiabetesPedigreeFunction	29.0	3.78
Age	9.0	1.17

Outlier Stats (Insulin columns missing values was re-imputed)

	no_of_outliers	no_of_outliers_percent
Pregnancies	4.0	0.52
Glucose	0.0	0.00
BloodPressure	14.0	1.82
SkinThickness	87.0	11.33
Insulin	9.0	1.17
BMI	8.0	1.04
DiabetesPedigreeFunction	29.0	3.78
Age	9.0	1.17

Outliers Treatment

```
Column in consideration: Pregnancies
Current number of rows: 768
Rows removed: 4
Rows removed (in %): 0.52


Column in consideration: Glucose
Current number of rows: 764
Rows removed: 0
Rows removed (in %): 0.0


Column in consideration: BloodPressure
Current number of rows: 764
Rows removed: 17
Rows removed (in %): 2.23


Column in consideration: SkinThickness
Current number of rows: 747
Rows removed: 85
Rows removed (in %): 11.38
```

```
Column in consideration: Insulin
Current number of rows: 662
Rows removed: 7
Rows removed (in %): 1.06


Column in consideration: BMI
Current number of rows: 655
Rows removed: 6
Rows removed (in %): 0.92


Column in consideration: DiabetesPedigreeFunction
Current number of rows: 649
Rows removed: 27
Rows removed (in %): 4.16


Column in consideration: Age
Current number of rows: 622
Rows removed: 9
Rows removed (in %): 1.45
```

Outliers Analysis Summary

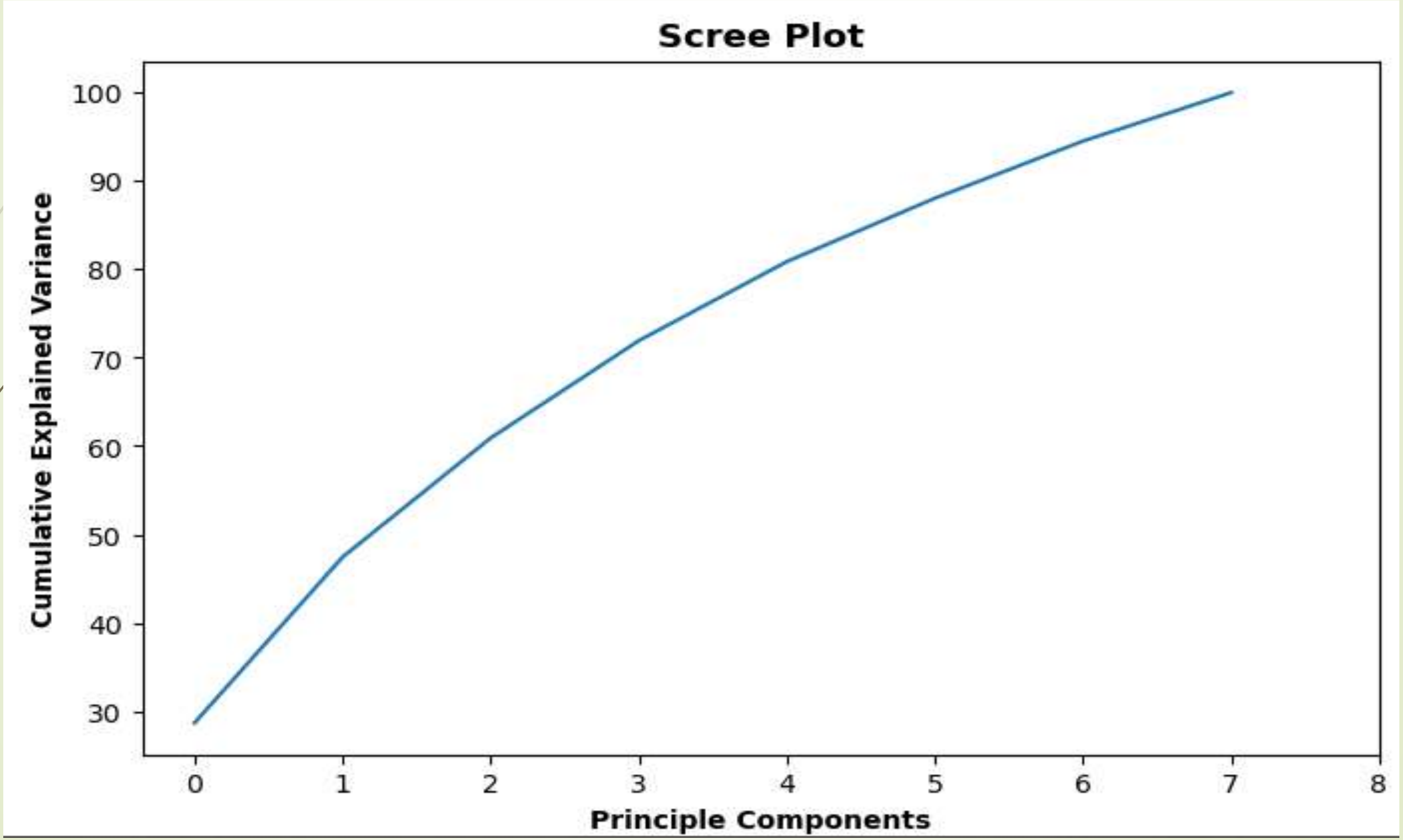
- It was found the **except Glucose** column, all the other independent variables had outliers present.
- Also, during reinspection of outliers (via values), it was observed that **Insulin** column had invariably **large number of outliers** present.
- So this **Insulin** column was **re-imputed** with mean and standard deviation values. The appropriate value was settled at **mean + standard deviation**, as this gave minimum number of outliers.
- Therefore, the outliers were removed by keeping only the values within **25th** and **75th percentile ± 1.5 times IQR** values of the respective column

Data Scaling and Principle Component Analysis

Scaling and PCA Summary

- Dataset was splitted in training and testing set, in **85% and 15% ratio** respectively. Random state 48 was used for results reproducibility.
- Splitted data was scaled using **MinMax Scaler**, because not all columns follow normal distribution
- Principle Component Analysis (PCA) was also performed in the dataset. It was observed that about **94% of the explained variance** was captured by **7 columns**, whereas the dataset has 8 columns (independent features)
- Therefore, in this particular dataset, the principle components aren't useful because there is **no significant reduction** of features observed.

PCA Summary – Scree Plot



Predictive Modelling Analysis

Predictive Model: Thought Process

- Since this is a Supervised Classification problem, Classification Machine Learning Algorithms were used.
- As the Outcome (Target) variable is not balanced, tree models would be a good suit as the primary intuition or maybe ensemble models.
- However, it is to good to test this dataset with DABL package, which can give us a rough estimate on how the dataset will perform on different models. From this baseline, ideas can be followed up to select the best model further.

Predictive Modelling: DABL Output

```
In [96]: ref_model = dabl.SimpleClassifier(random_state=0).fit(diabetes_data_missing_imputed_outlier_removed, target_col="Outcome")
         ref_model
```

```
Best model:
LogisticRegression(C=0.1, class_weight='balanced', max_iter=1000)
Best Scores:
accuracy: 0.736 average_precision: 0.706 roc_auc: 0.828 recall_macro: 0.726 f1_macro: 0.713
```

```
Out[96]:
SimpleClassifier
SimpleClassifier(random_state=0)
```

Predictive Modelling: Different Classifiers Output

```
# Running the data with LogisticRegression

clf = LogisticRegression(C=0.1, class_weight='balanced', max_iter=1000)
model, train_acc, test_acc = ML_model_classifier(clf, X_train, X_test, y_train, y_test, verbose = 1)
```

```
Model: LogisticRegression(C=0.1, class_weight='balanced', max_iter=1000)
Training Accuracy: 72.74%
Test Accuracy: 80.43%
```

```
# Running the data with DecisionTreeClassifier

clf = DecisionTreeClassifier(class_weight='balanced')
model, train_acc, test_acc = ML_model_classifier(clf, X_train, X_test, y_train, y_test, verbose = 1)
```

```
Model: DecisionTreeClassifier(class_weight='balanced')
Training Accuracy: 100.0%
Test Accuracy: 71.74%
```

Predictive Modelling: Different Classifiers Output

```
# Running the data with RandomForestClassifier

clf = RandomForestClassifier(n_estimators=200, class_weight='balanced')
model, train_acc, test_acc = ML_model_classifier(clf, X_train, X_test, y_train, y_test, verbose = 1)
```

```
Model: RandomForestClassifier(class_weight='balanced', n_estimators=200)
Training Accuracy: 100.0%
Test Accuracy: 81.52%
```

```
# Running the data with KNeighborsClassifier

clf = KNeighborsClassifier()
model, train_acc, test_acc = ML_model_classifier(clf, X_train, X_test, y_train, y_test, verbose = 1)
```

```
Model: KNeighborsClassifier()
Training Accuracy: 83.49%
Test Accuracy: 77.17%
```

Predictive Modelling: Different Classifiers Output

```
# Running the data with SVC

clf = SVC(kernel = 'linear',gamma = 'scale', shrinking = False)
model, train_acc, test_acc = ML_model_classifier(clf, X_train, X_test, y_train, y_test, verbose = 1)
```

```
Model: SVC(kernel='linear', shrinking=False)
Training Accuracy: 77.74%
Test Accuracy: 77.17%
```

```
# Running the data with XGBoost Classifier

xgb_classifier = xgb.XGBClassifier()
model, train_acc, test_acc = ML_model_classifier(xgb_classifier, X_train, X_test, y_train, y_test, verbose = 1)
```

```
Model: XGBClassifier(base_score=None, booster=None, colsample_bylevel=None,
              colsample_bynode=None, colsample_bytree=None,
              enable_categorical=False, gamma=None, gpu_id=None,
              importance_type=None, interaction_constraints=None,
              learning_rate=None, max_delta_step=None, max_depth=None,
              min_child_weight=None, missing=nan, monotone_constraints=None,
              n_estimators=100, n_jobs=None, num_parallel_tree=None,
              predictor=None, random_state=None, reg_alpha=None,
              reg_lambda=None, scale_pos_weight=None, subsample=None,
              tree_method=None, validate_parameters=None, verbosity=None)
Training Accuracy: 77.74%
Test Accuracy: 77.17%
```

Predictive Model: Initial Run

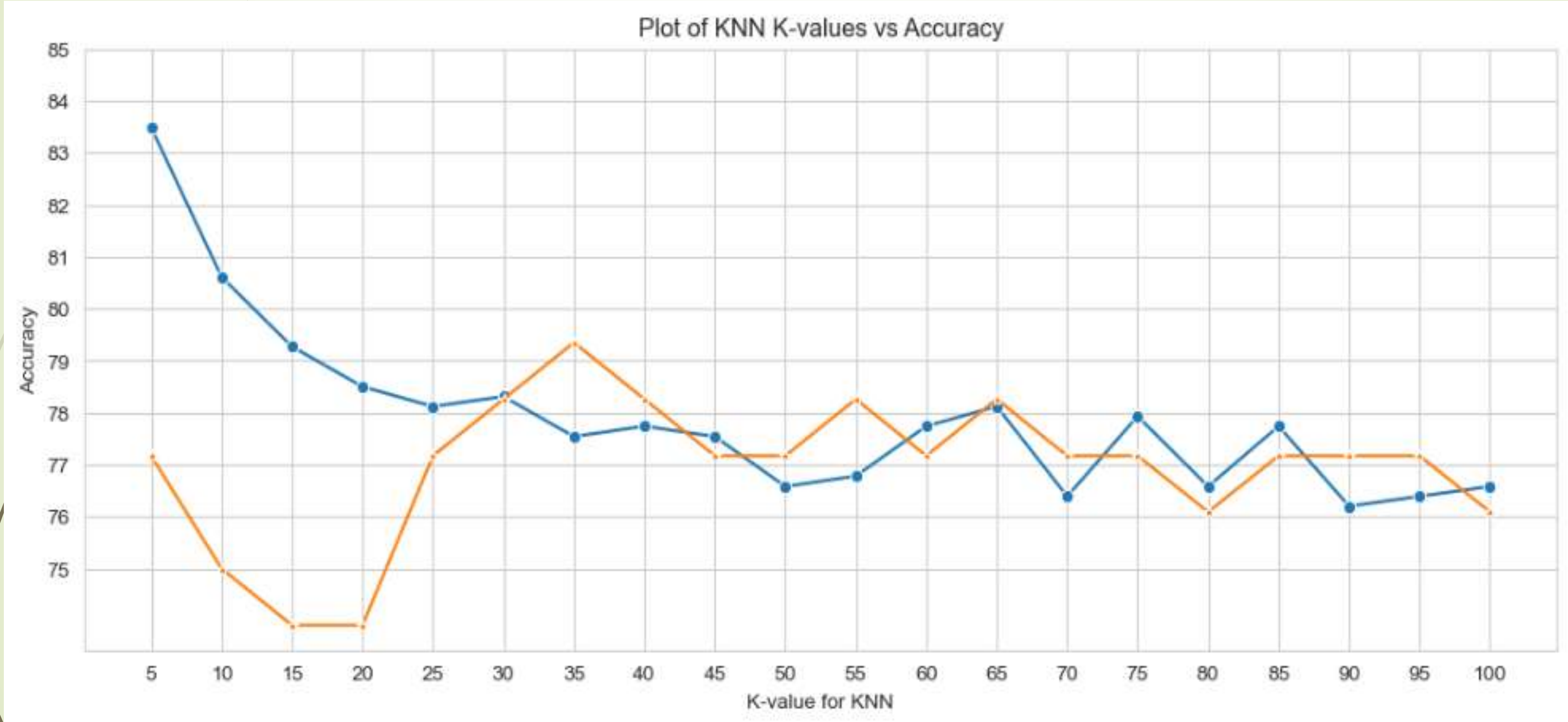
- Running the dataset with DABL package gave a LogisticRegression as a baseline model to start with. However, DecisionTreeClassifier was nominated as the best model for dataset with principle components.
- LogisticRegression, DecisionTreeClassifier, RandomForestClassifier, SVC, XGBClassifier and KNeighborsClassifier were implemented one by one and training & testing accuracies were compared.
- It was found that DecisionTreeClassifier, RandomForestClassifier were overfitting (Training Accuracy:**100%**, Testing accuracy at about **70% to 80%)**

Predictive Model: Initial Run

- LogisticRegression model underfits. (Training Accuracy: **72.74%**, Test Accuracy: **80.43%**)
- KNeighborsClassifier gave best accuracy on training data with relatively lower on testing data (Training Accuracy: **83.49%**, Test Accuracy: **77.17%**)
- SVC and XGBClassifier are the best classification models for this problem, and gave highest accuracies of **77%** in both training and testing data with no overfitting.

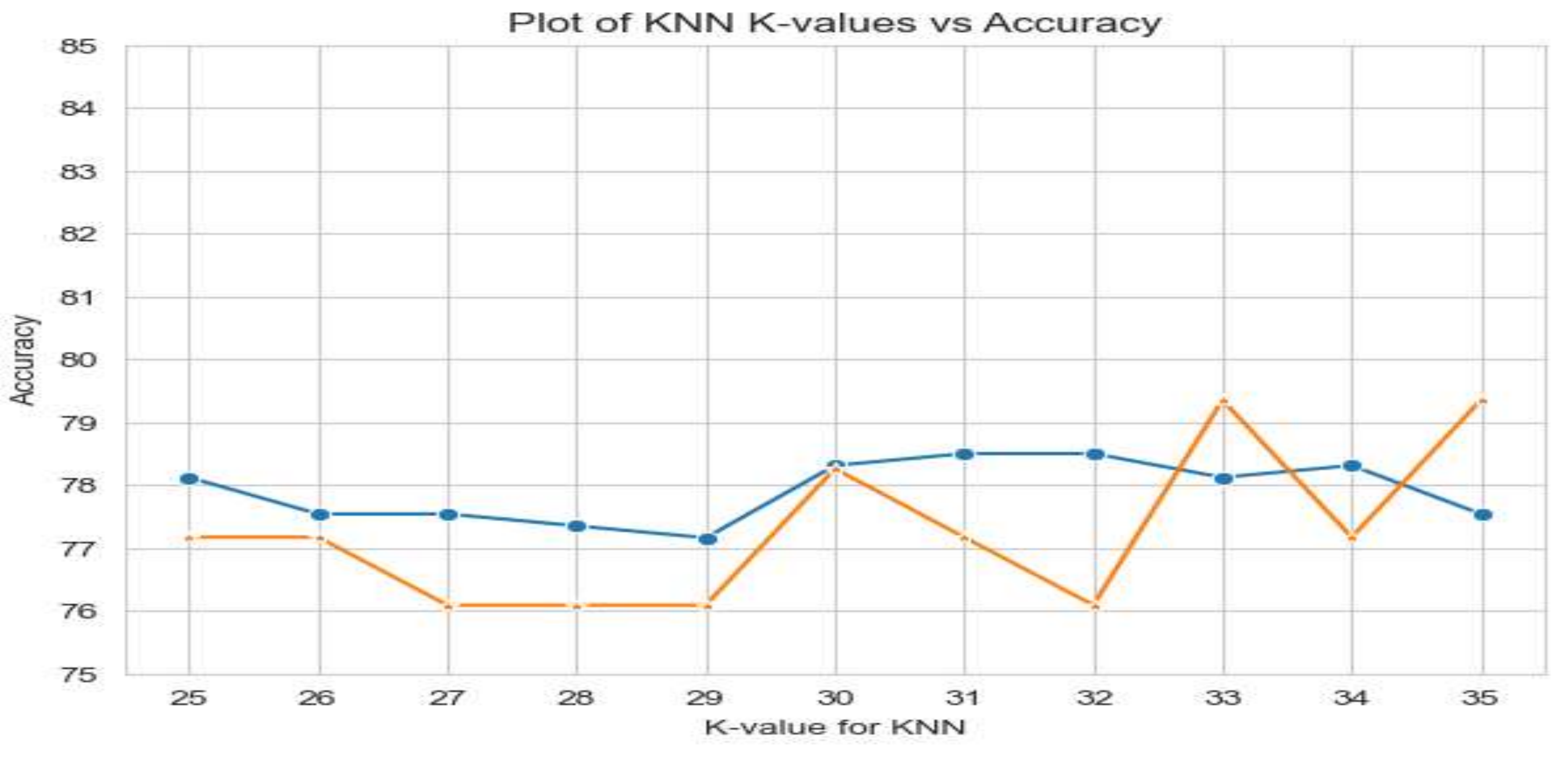
Predictive Modelling Analysis: KNN Classifier

Predictive Model: Finding best K



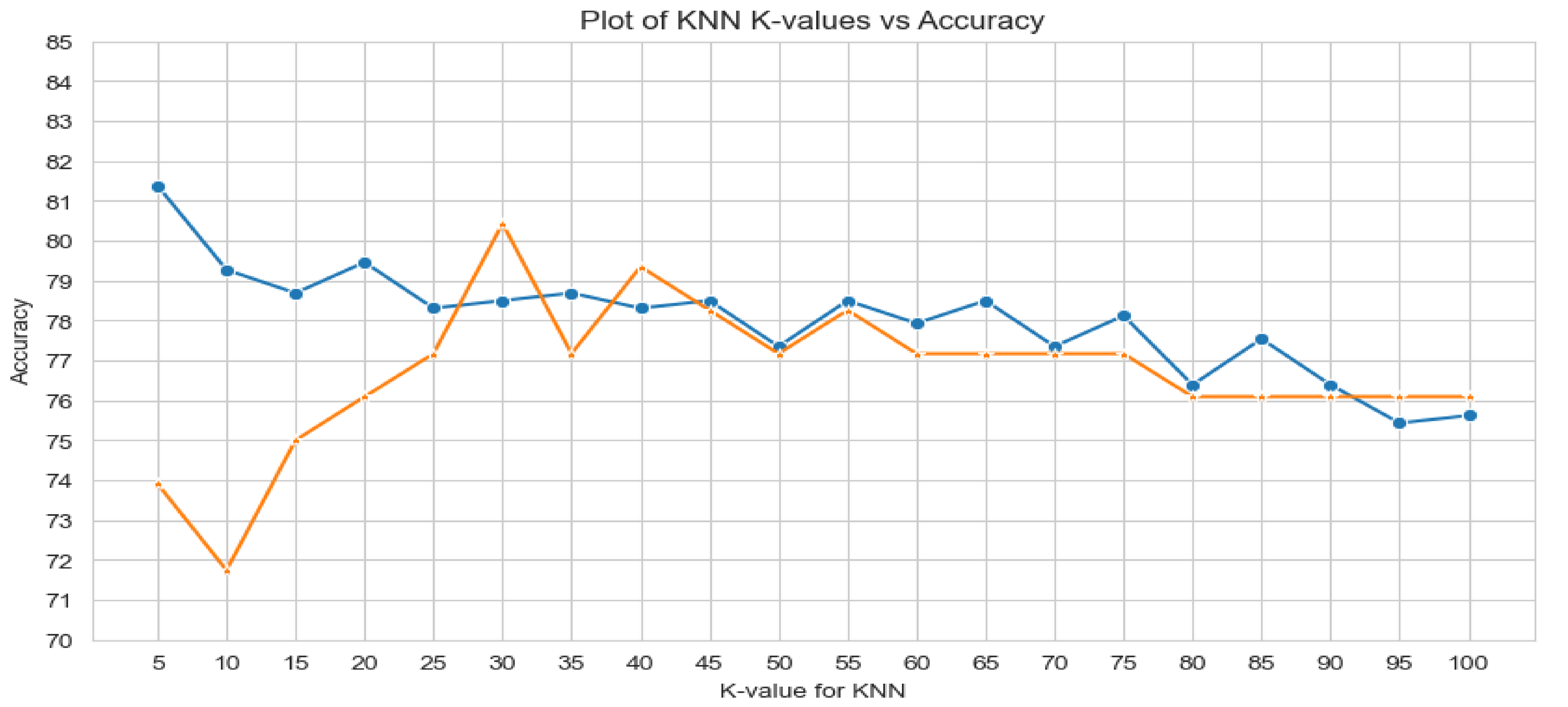
Predictive Model: Finding Best K

Zoomed the k value between 25 to 35



Predictive Model: Finding Best K

For Principle Components dataset



Predictive Model: Tuning KNN Model

- Since KNN model gave highest accuracy in training data, this was optimized for best K value using loops and graphs. (even though the highest accuracy on training data is given by random forest and decision tress, however, those model are overfitting)
- It was found that when **k=30**, the model gave its best accuracy on training and testing data of about **78%** without overfitting.
- KNN model was also checked for its best values using **7 principle components**, however, the training and testing accuracies of about **78% converged** at **k = 45**. Moreover, there was **no significant dimensionality reduction achieved with PCA**, therefore it is disregarded in further metric calculations.

Predictive Modelling Analysis: Random Forest Classifier

Predictive Model: RandomForestClassifier

```
grid_search.best_score_
```

```
0.7851596516690856
```

```
best_parameters = grid_search.best_params_
print(best_parameters)
```

```
{'max_depth': 100, 'min_samples_leaf': 15, 'n_estimators': 700}
```

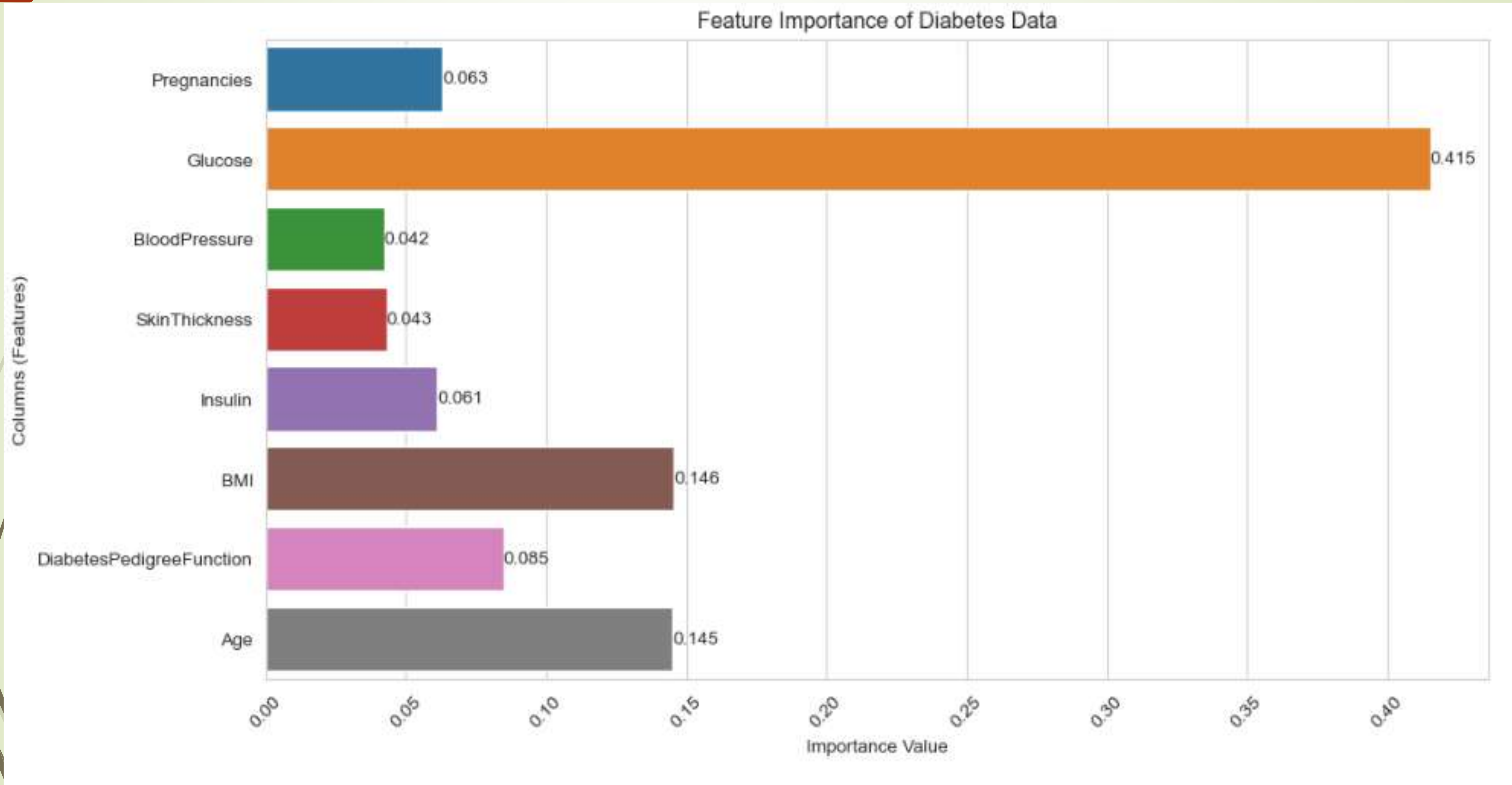
```
rf_best = grid_search.best_estimator_
rf_best
```

```
RandomForestClassifier
RandomForestClassifier(max_depth=100, min_samples_leaf=15, n_estimators=700,
                       n_jobs=-1)
```

```
print(f'Training Accuracy: {round((accuracy_score(y_train, y_pred_train)*100),2)}%')
print(f'Test Accuracy: {round((accuracy_score(y_test, y_pred_test)*100),2)}%')
```

```
Training Accuracy: 81.77%
Test Accuracy: 73.91%
```

Predictive Model: Feature Importance

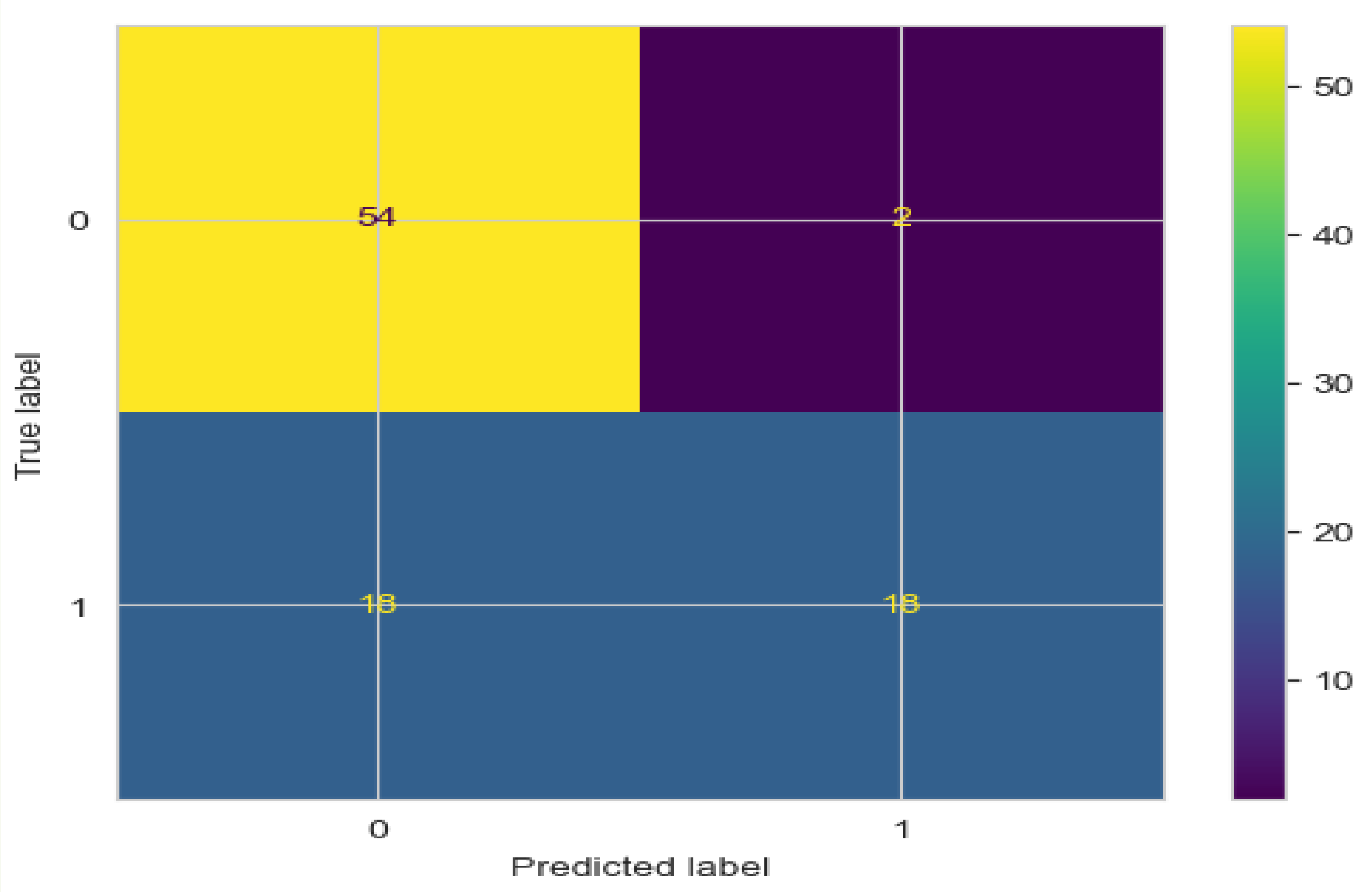


Predictive Model: Random Forest Summary

- GridSearchCV was used to find optimum parameters for RandomForestClassifier
- GridSearchCV returned the tuned RandomForestClassifier model which has Training Accuracy: **81.19%** Test Accuracy: **73.91%**
- This model even though has best Training accuracy but **still it is overfitting.**
- Feature importance was extracted from Random Forest and it was identified that '**Glucose**' is the most important feature influencing the target, followed by '**Age**' and '**BMI**'

Predictive Modelling: Best Model's (KNN Classifier) Performance Metrics Analysis

Performance Metrics: Confusion Matrix



Performance Metrics: Confusion Matrix

```
tn, fp, fn, tp = confusion_matrix(y_test, y_pred_test, labels=model.classes_).ravel()
specificity = tn / (tn+fp)
sensitivity = tp / (tp+fn)
```

```
print(f"True Negative: {tn}")
print(f"False Positive: {fp}")
print(f"False Negative: {fn}")
print(f"True Positive: {tp}")
```

```
True Negative: 54
False Positive: 2
False Negative: 18
True Positive: 18
```

Performance Metrics: Classification Report

```
print(classification_report(y_test, y_pred_test))
```

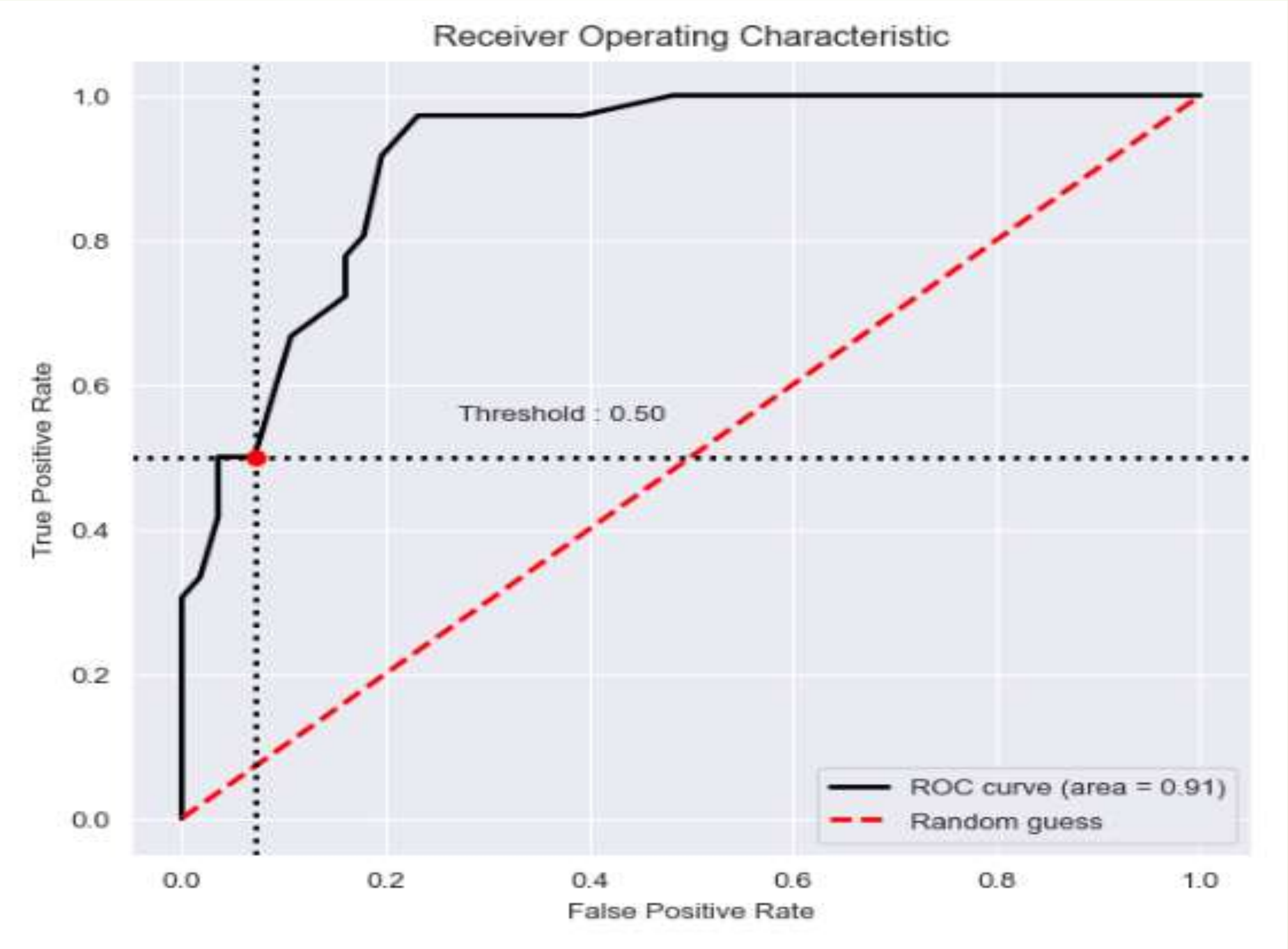
	precision	recall	f1-score	support
0	0.75	0.96	0.84	56
1	0.90	0.50	0.64	36
accuracy			0.78	92
macro avg	0.82	0.73	0.74	92
weighted avg	0.81	0.78	0.77	92

Performance Metrics: Sensitivity, Specificity

```
print(f"Sensitivity/Recall: {round(sensitivity,2)}")
print(f"Specificity: {round(specificity,2)}")
```

```
Sensitivity/Recall: 0.5
Specificity: 0.96
```

Performance Metrics: ROC Curve



Performance Metrics: Cross Validation Score

```
k_folds = KFold(n_splits = 10)

scores = cross_val_score(model, X, y, cv = k_folds)
```

```
CPU times: total: 234 ms
Wall time: 226 ms
```

```
print("Cross Validation Scores: \n", scores)
print(f"\nAverage CV Score: {round(scores.mean()*100,2)}%")
print("\nNumber of CV Scores used in Average: ", len(scores))
```

```
Cross Validation Scores:
 [0.69354839 0.80645161 0.72580645 0.60655738 0.70491803 0.73770492
 0.80327869 0.80327869 0.75409836 0.78688525]

Average CV Score: 74.23%

Number of CV Scores used in Average:  10
```

Performance Metrics: Test Predictions

Task II (I): Predicition of test datapoints and comparison with actual datapoints

```
# since predicted test values were in numpy array type, converting it to series type

y_pred_test = pd.Series(y_pred_test, index=y_test.index)
```

```
type(y_test), type(y_pred_test)
```

```
(pandas.core.series.Series, pandas.core.series.Series)
```

```
# preparing a dataframe to get th actual values and predicted values side by side

actual_pred_comparison = pd.DataFrame([y_test, y_pred_test], index=['actual_outcomes', 'predicted_outcomes']).T
actual_pred_comparison.head(6)
```

	actual_outcomes	predicted_outcomes
0	1	1
121	0	0
325	0	0
214	1	0
651	0	0
345	0	0

```
# exporting the actual vs predicted values comparison dataframe as csv file (for reporting and submission purpose)

actual_pred_comparison.to_csv('actual_pred_comparison.csv', index = False)
```

actual_pred_com
parison.csv

Performance Metrics: Summary

- Overall, the final best model is KNN with **K=30**.
  - Training Accuracy: **78.31%**
  - Test Accuracy: **78.26%**
- Following are the confusion metrics obtained on test data:
  - True Negative: **54**
  - False Positive: **2**
  - False Negative: **18**
  - True Positive: **18**
  - Sensitivity/Recall: **0.5**
  - Specificity: **0.96**
  - ROC Curve Area: **0.91**

Tableau Report

Tableau Report: Overview

- "diabetes_data_for_tableau_report.csv" file is used for tableau report. Also attached with this slide.
- Link to report: https://public.tableau.com/app/profile/lavkush.singh4748/viz/PCDS-DataScienceCapstoneTableauReport/ProportionofDiabeticPopulation
- There are 16 sheets and 6 Dashboards where information is presented in terms of various visualizations.
- Dashboard Names:
  - Basic Diabetes Data Information
  - Diabetes Variables Relationship - I
  - Diabetes Variables Relationship - II
  - BMI and Blood Stats by Age
  - Various Parameters vs Pregnancies Count
  - Various Parameters vs Age

diabetes_data_for_tableau_report

Tableau Report: Snips (Sheets)

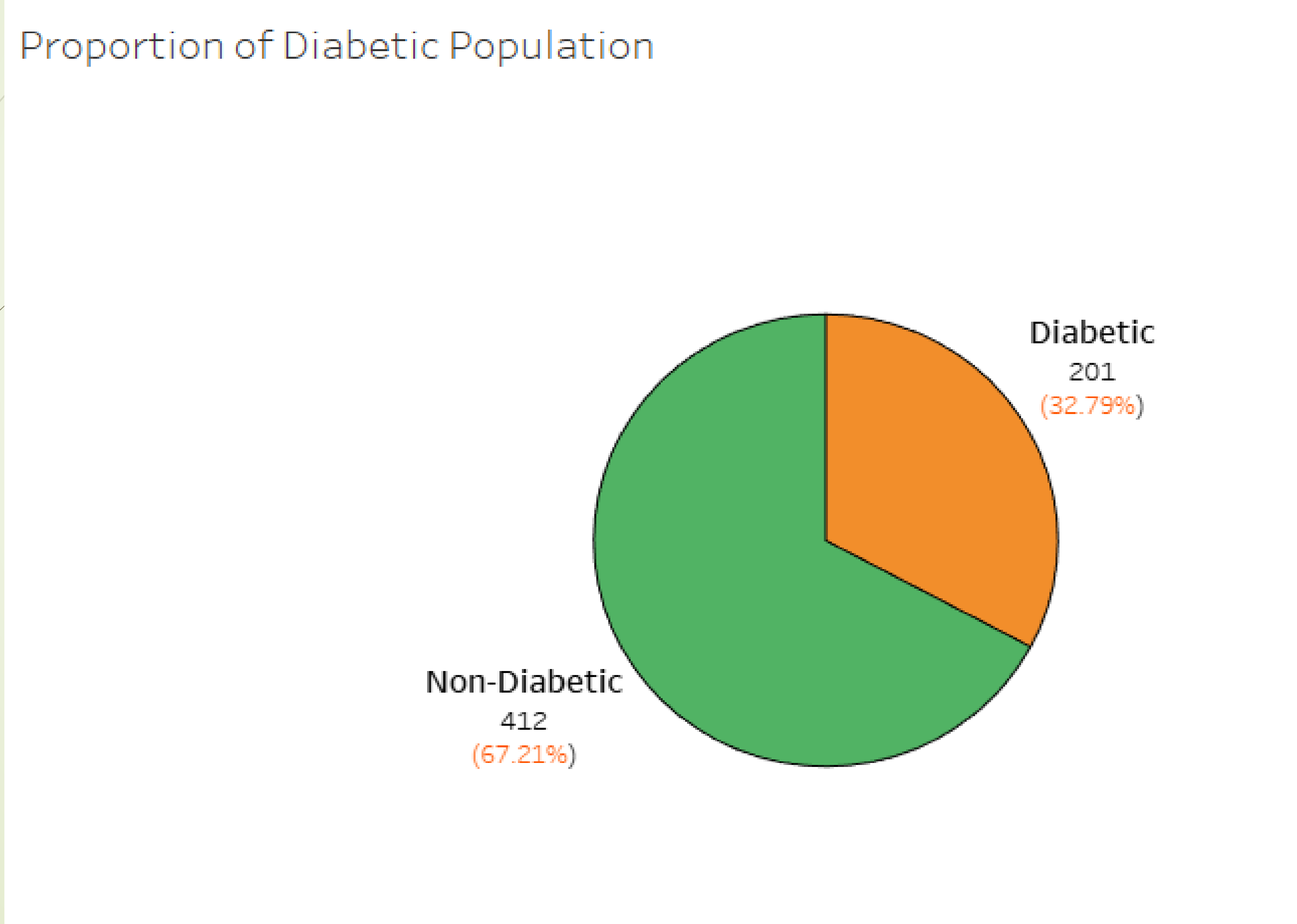


Tableau Report: Snips (Sheets)

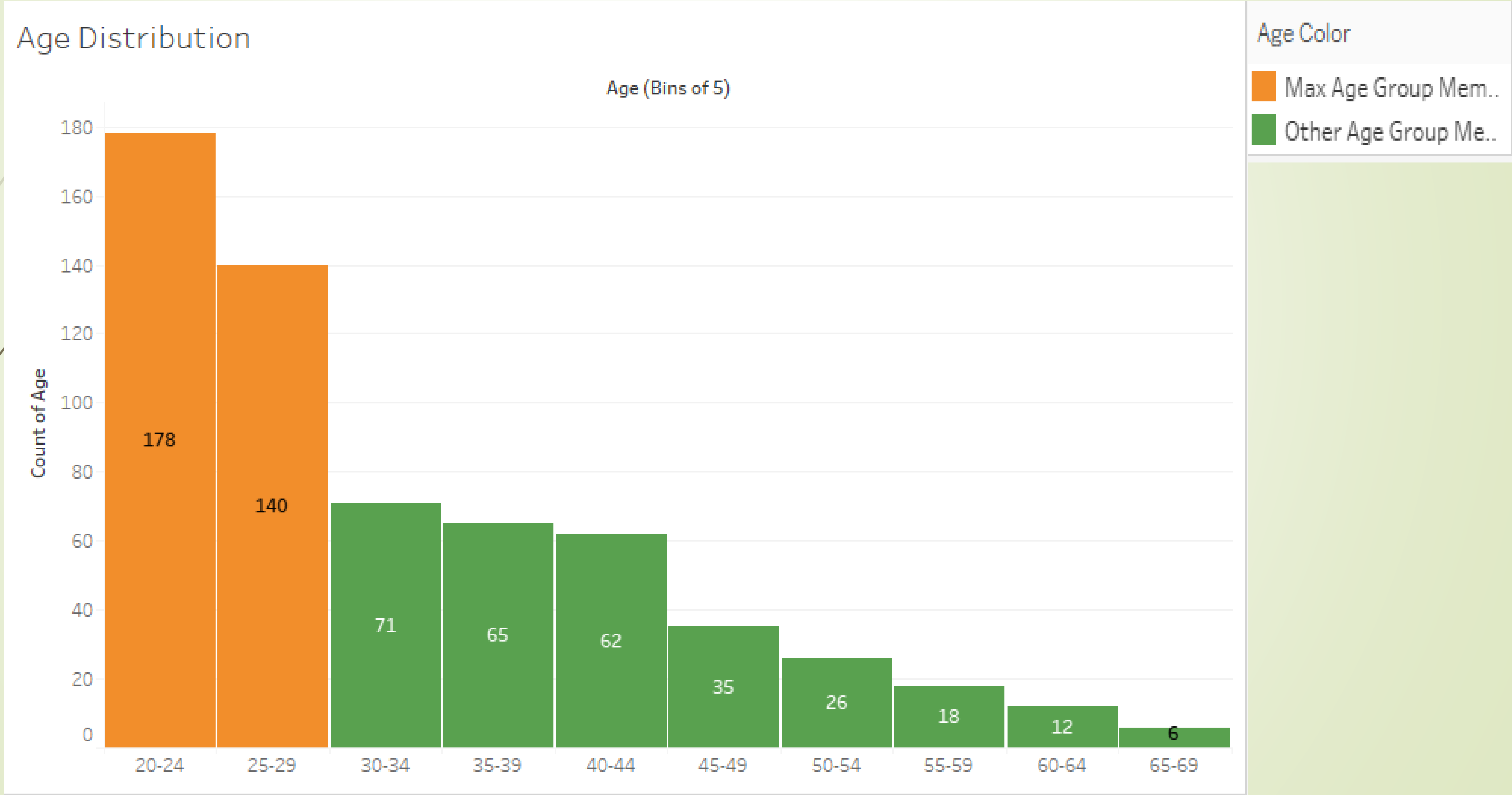


Tableau Report: Snips (Sheets)

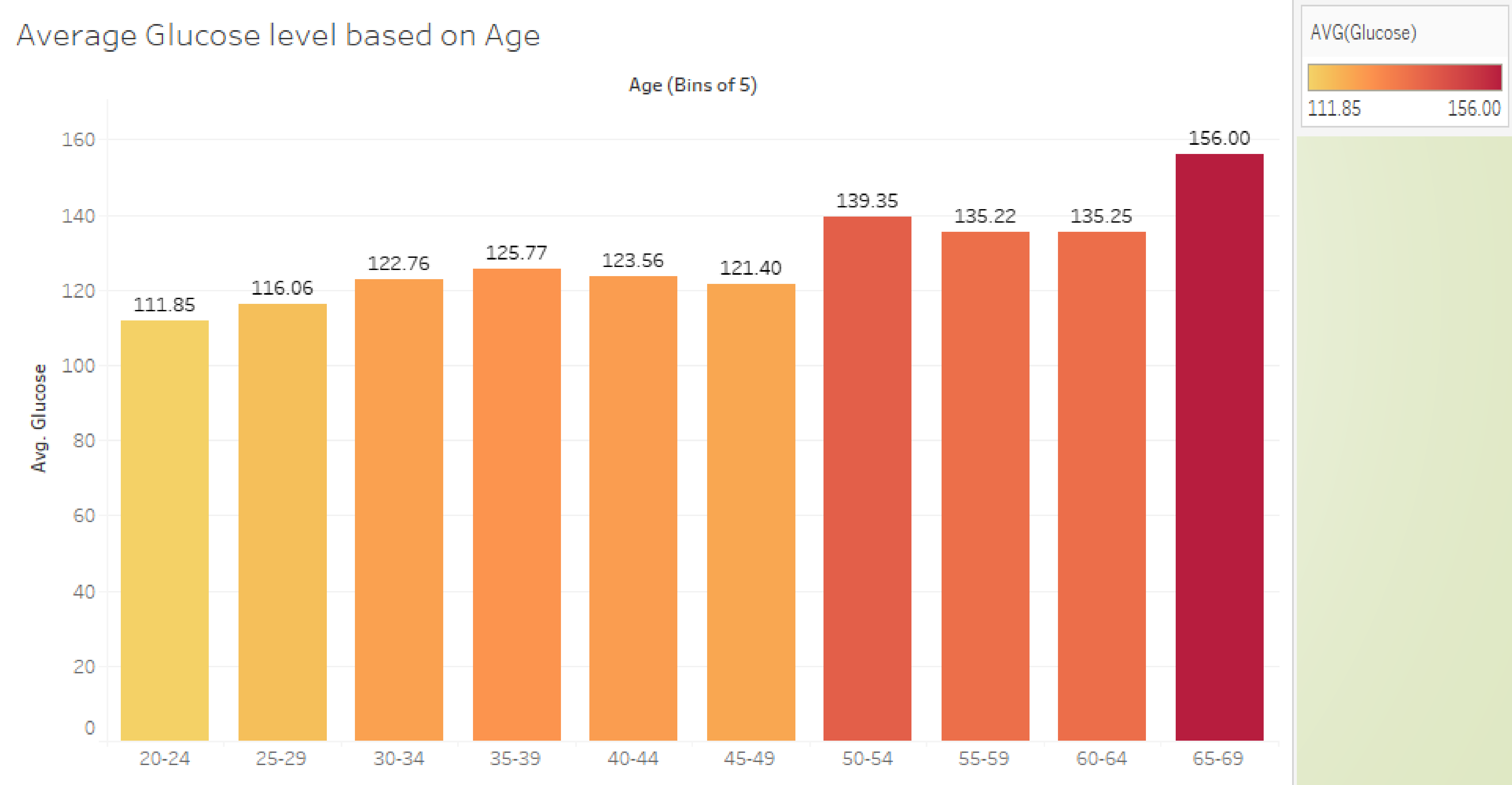


Tableau Report: Snips (Sheets)

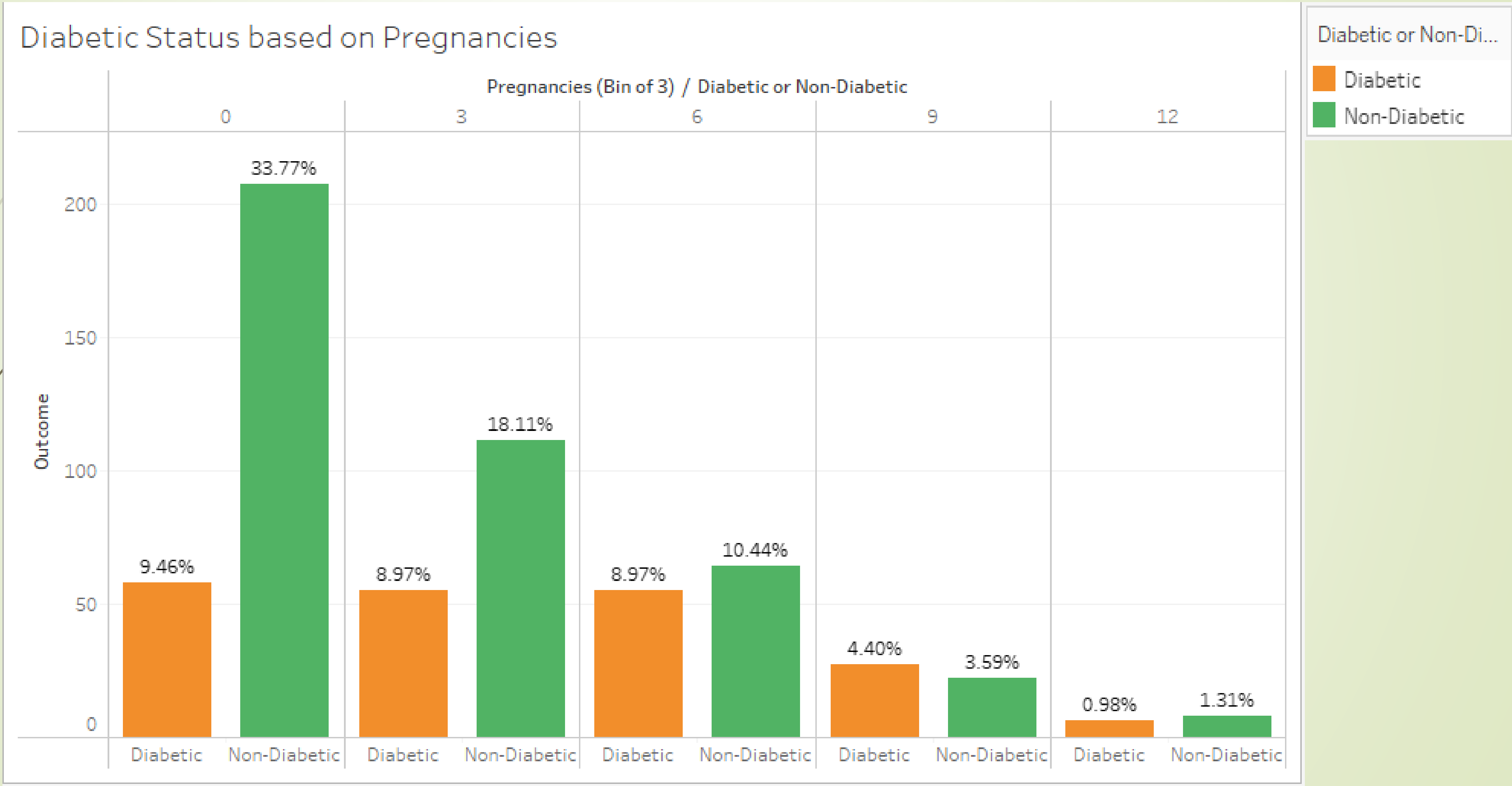


Tableau Report: Snips (Sheets)

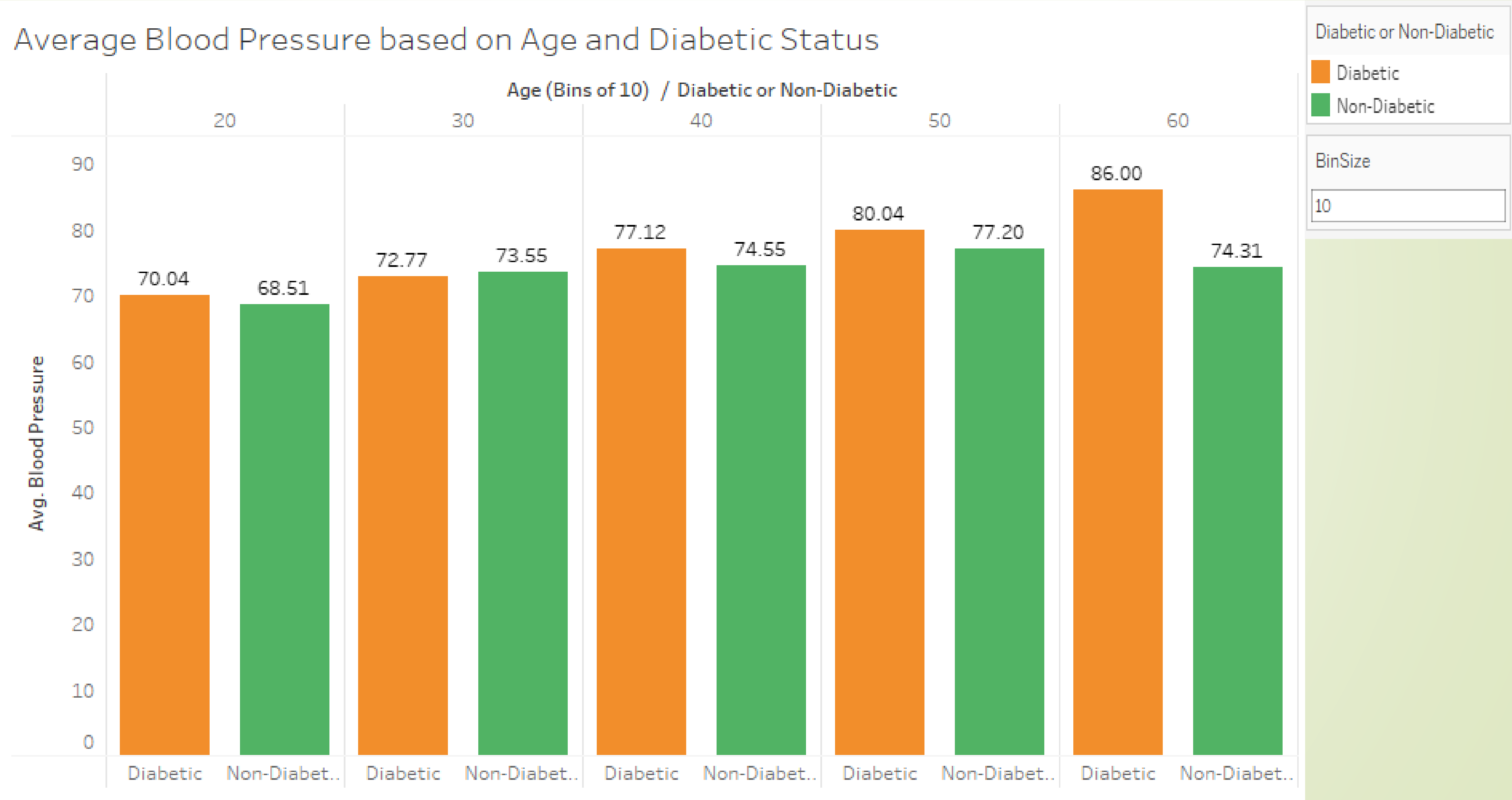


Tableau Report: Snips (Sheets)

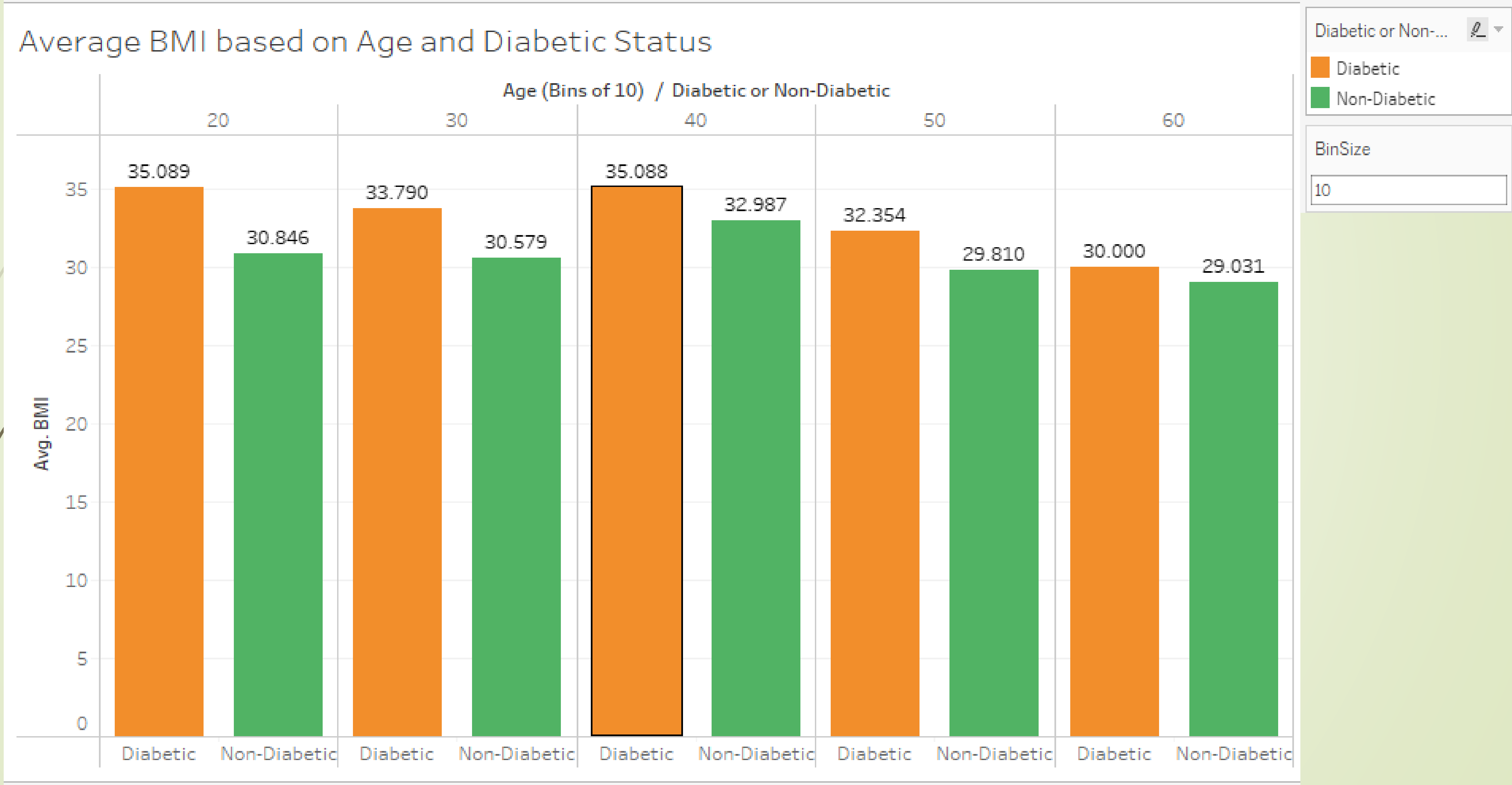


Tableau Report: Snips (Sheets)

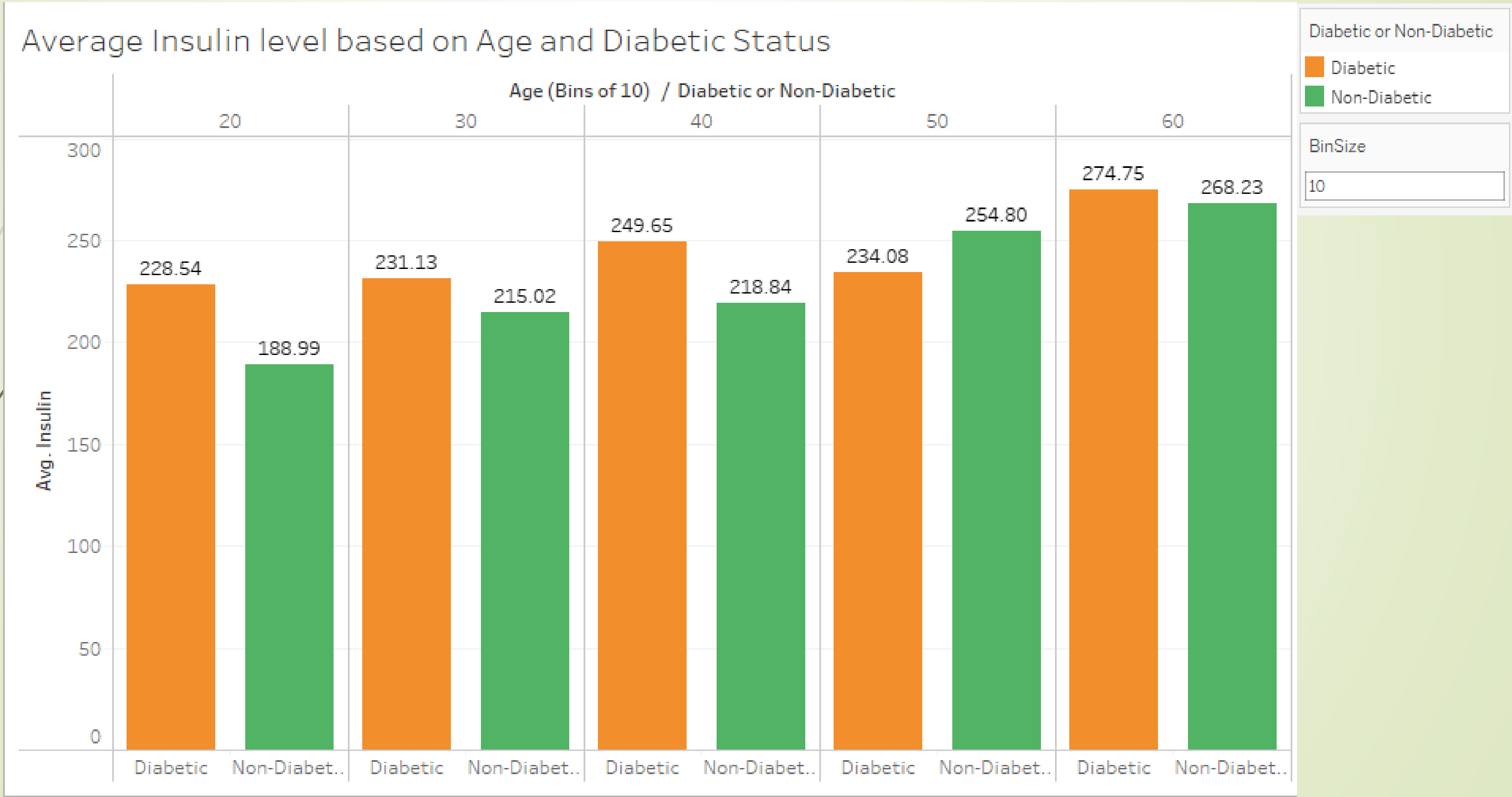


Tableau Report: Snips (Sheets)

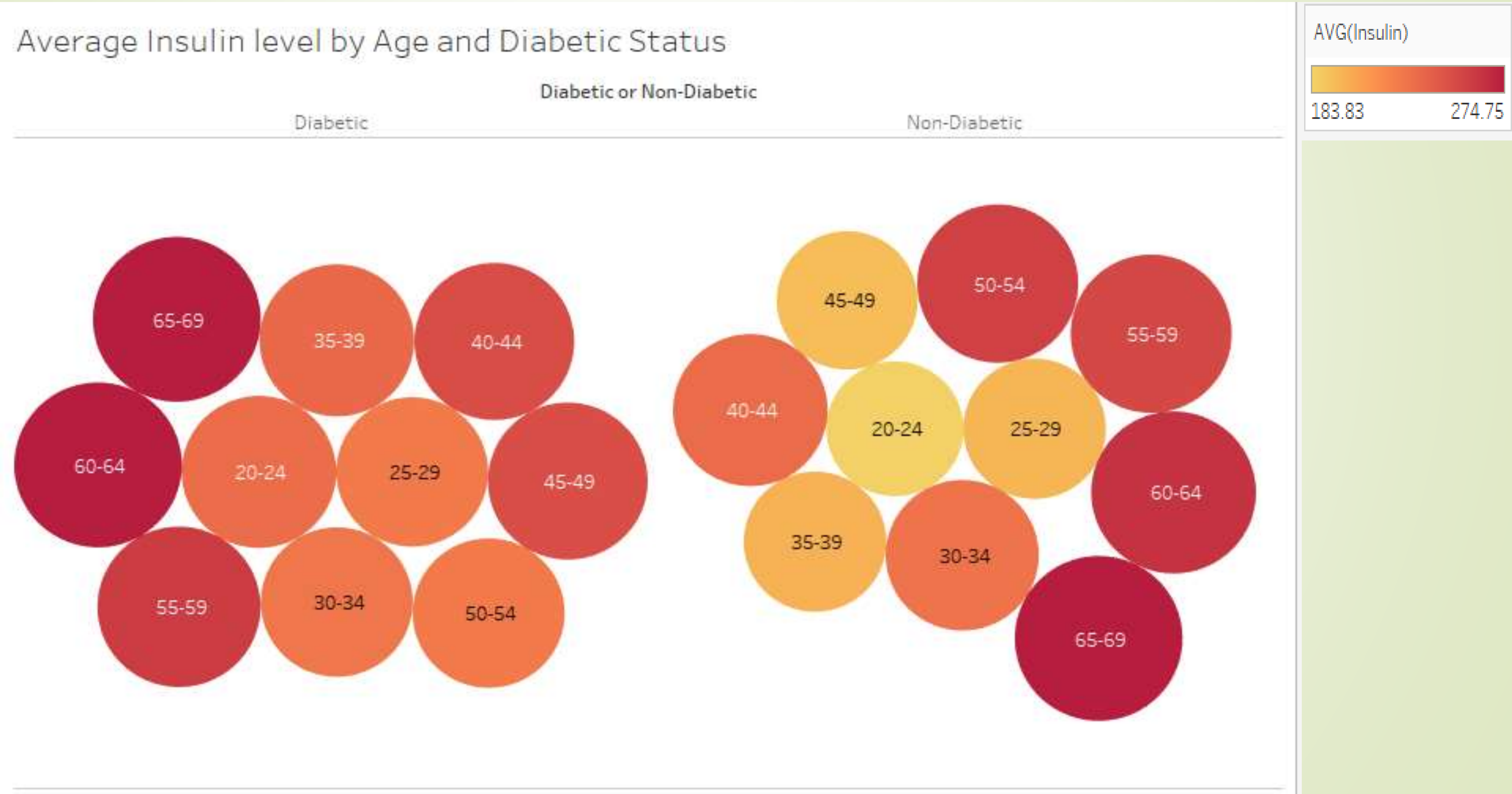


Tableau Report: Snips (Sheets)

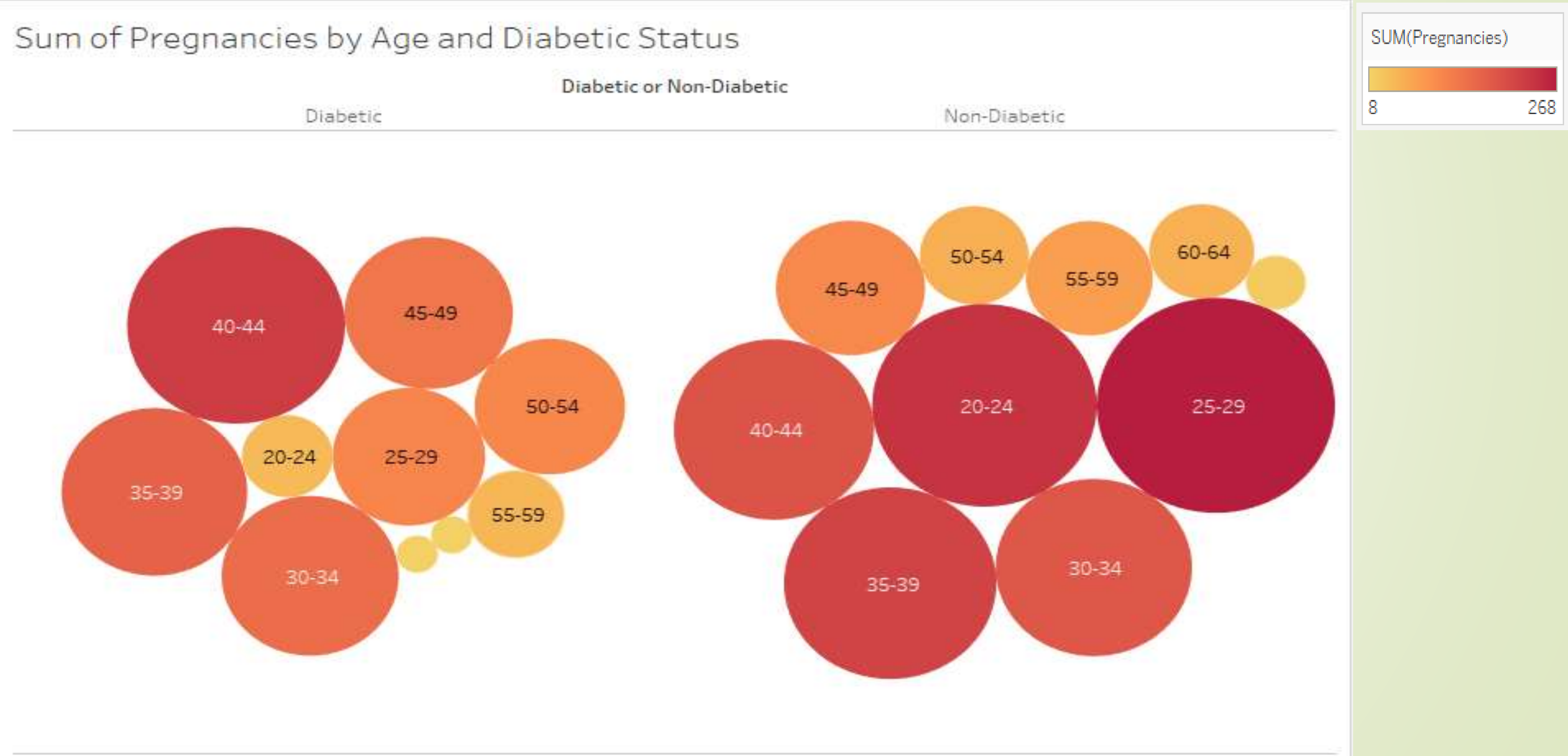


Tableau Report: Snips (Sheets)

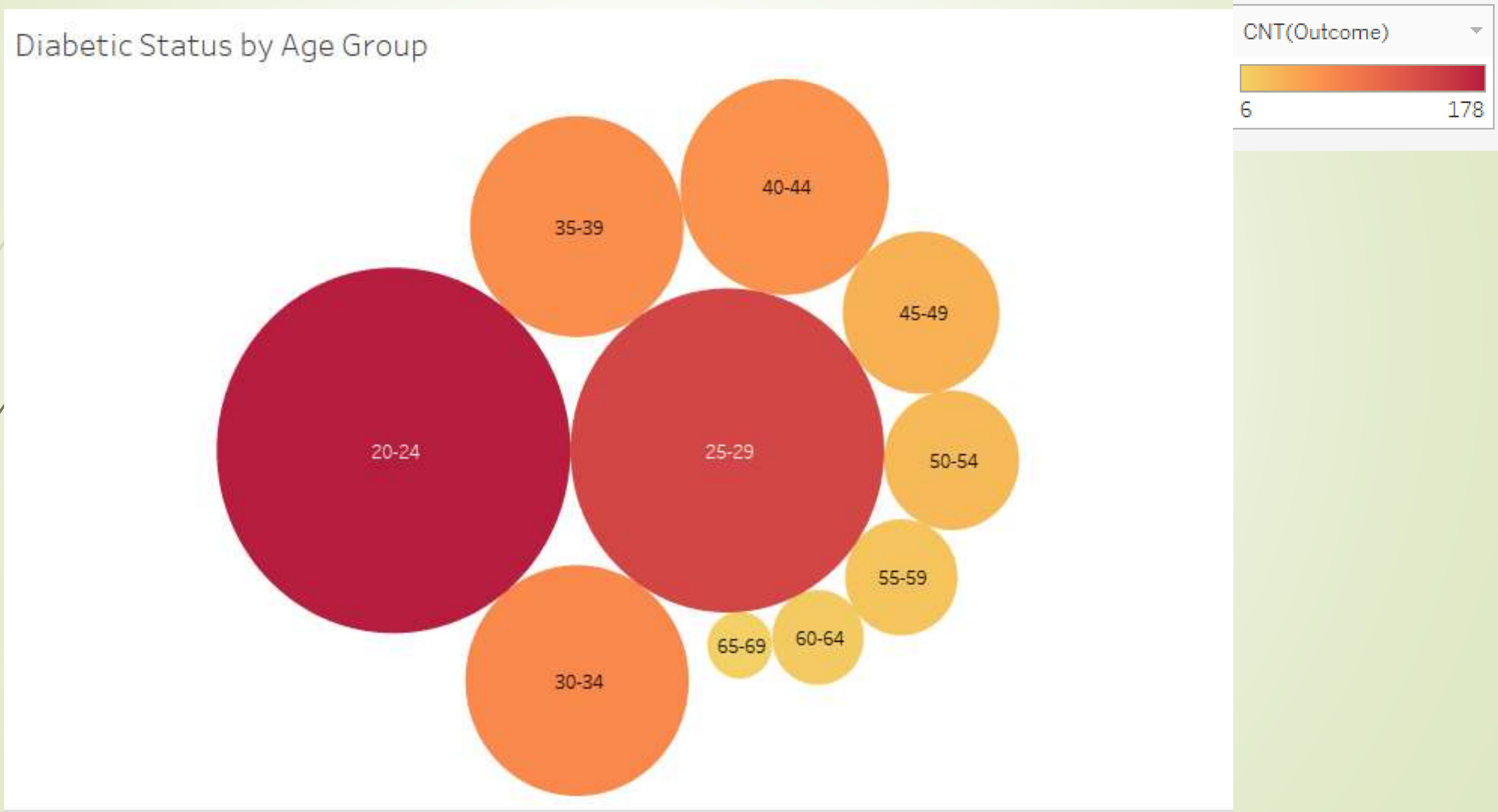


Tableau Report: Snips (Sheets)

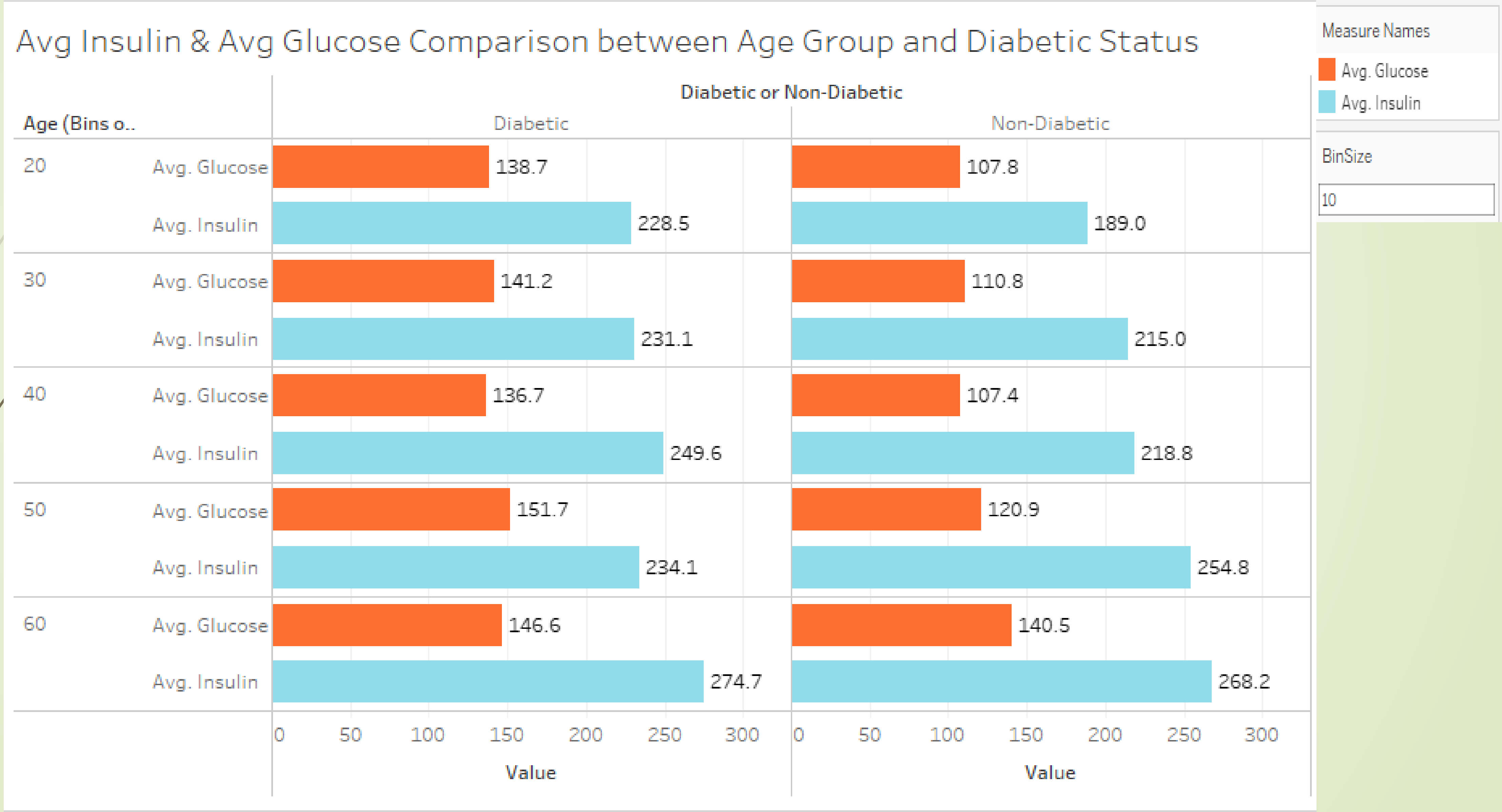


Tableau Report: Snips (Sheets)

Different Parameters based on Age

	Age (Bins of 5)									
	20-24	25-29	30-34	35-39	40-44	45-49	50-54	55-59	60-64	65-69
Avg. BMI	31.0	32.5	31.2	32.8	34.5	33.3	32.6	29.2	29.2	29.5
Avg. Blood Pressure	68.0	69.9	71.8	74.7	73.9	79.4	80.6	76.1	77.0	78.7
Avg. Diabetes Pedigree Function	0.4	0.4	0.5	0.4	0.4	0.4	0.5	0.5	0.4	0.4
Avg. Glucose	111.9	116.1	122.8	125.8	123.6	121.4	139.3	135.2	135.3	156.0
Avg. Insulin	190.5	204.4	228.3	215.4	242.2	222.3	235.3	255.4	267.7	274.7
Avg. Skin Thickness	27.0	29.1	28.8	30.6	30.3	30.4	28.5	28.6	27.3	29.1

Tableau Report: Snips (Sheets)

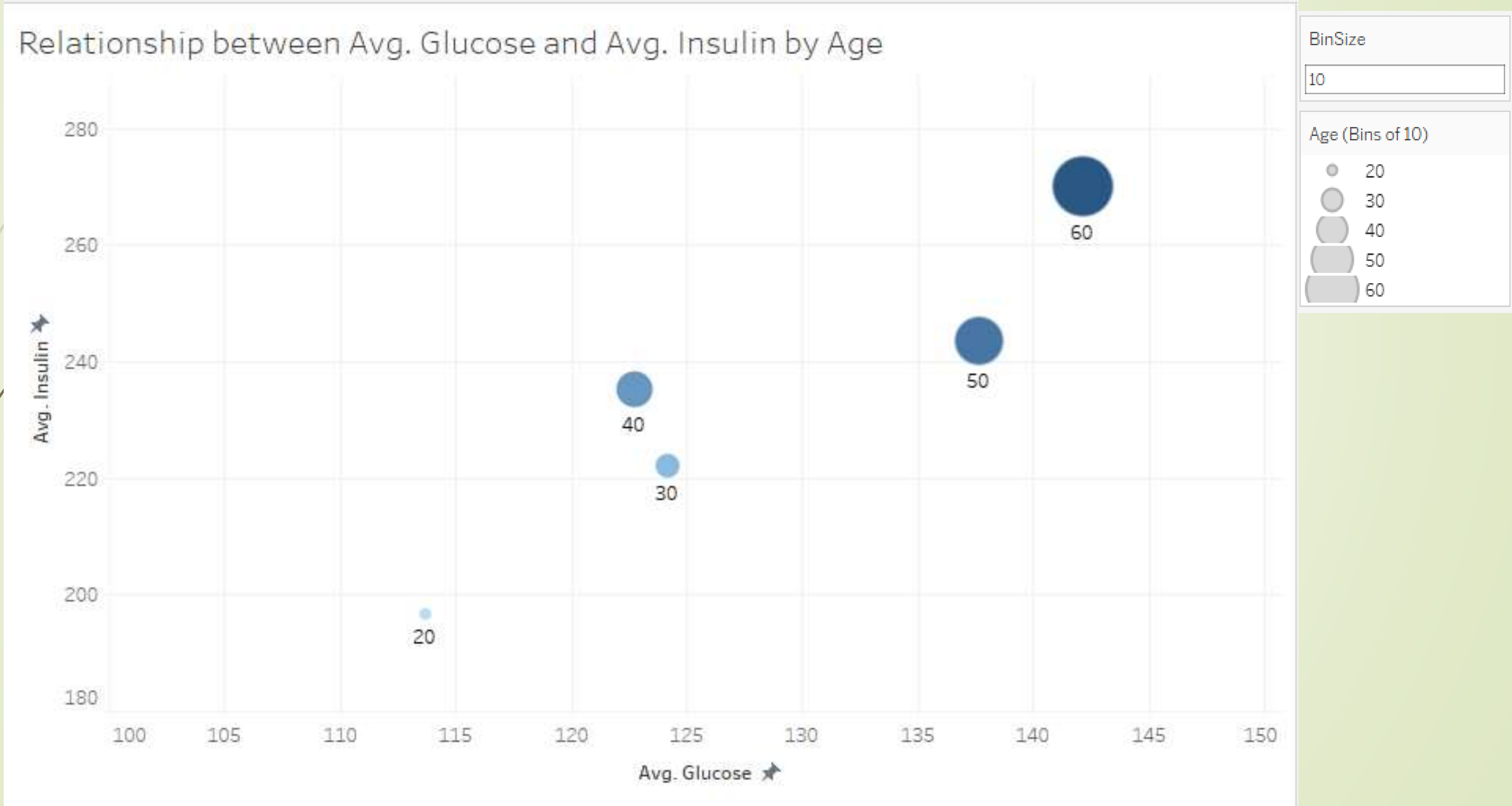


Tableau Report: Snips (Sheets)

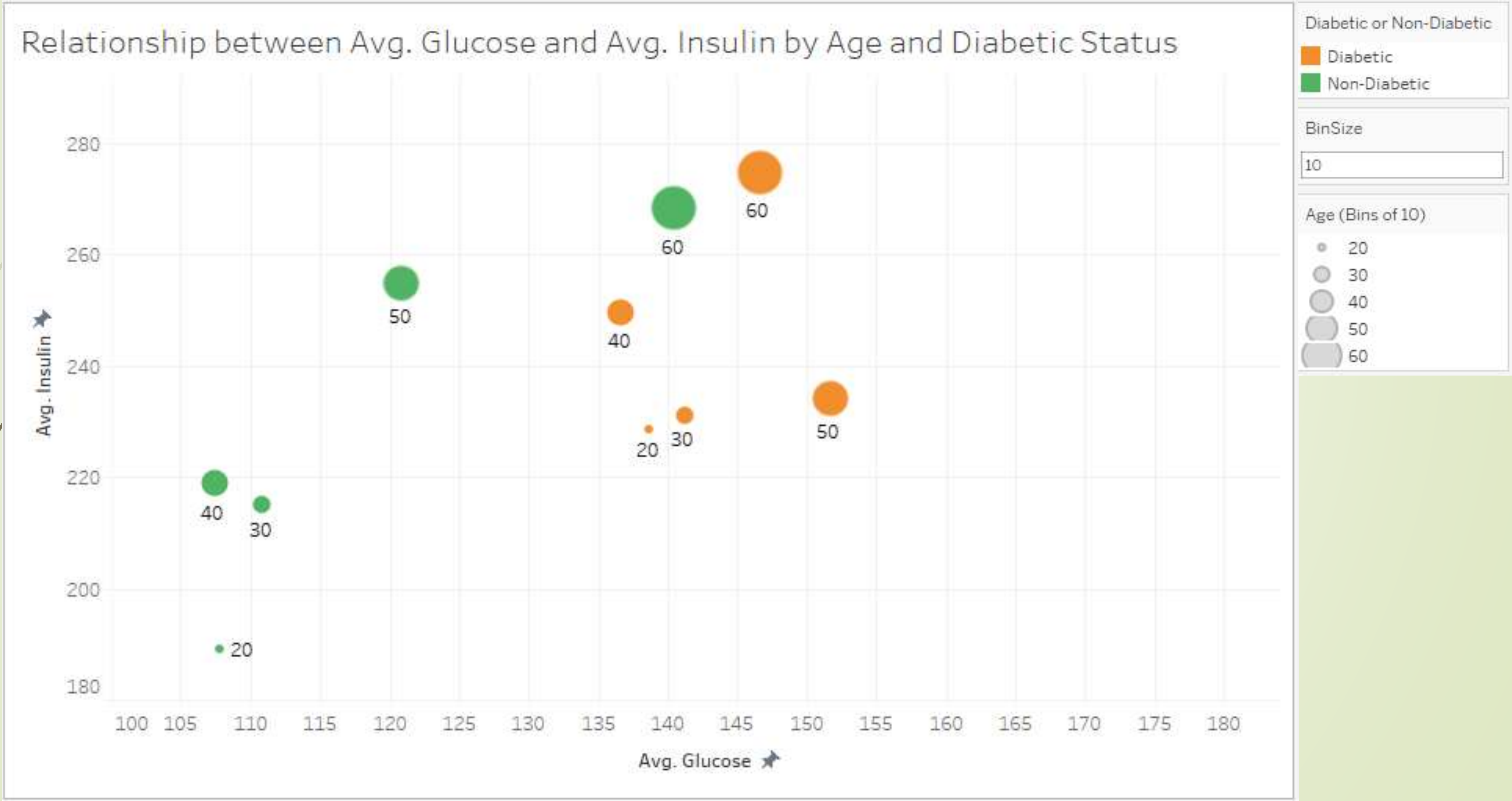


Tableau Report: Snips (Sheets)

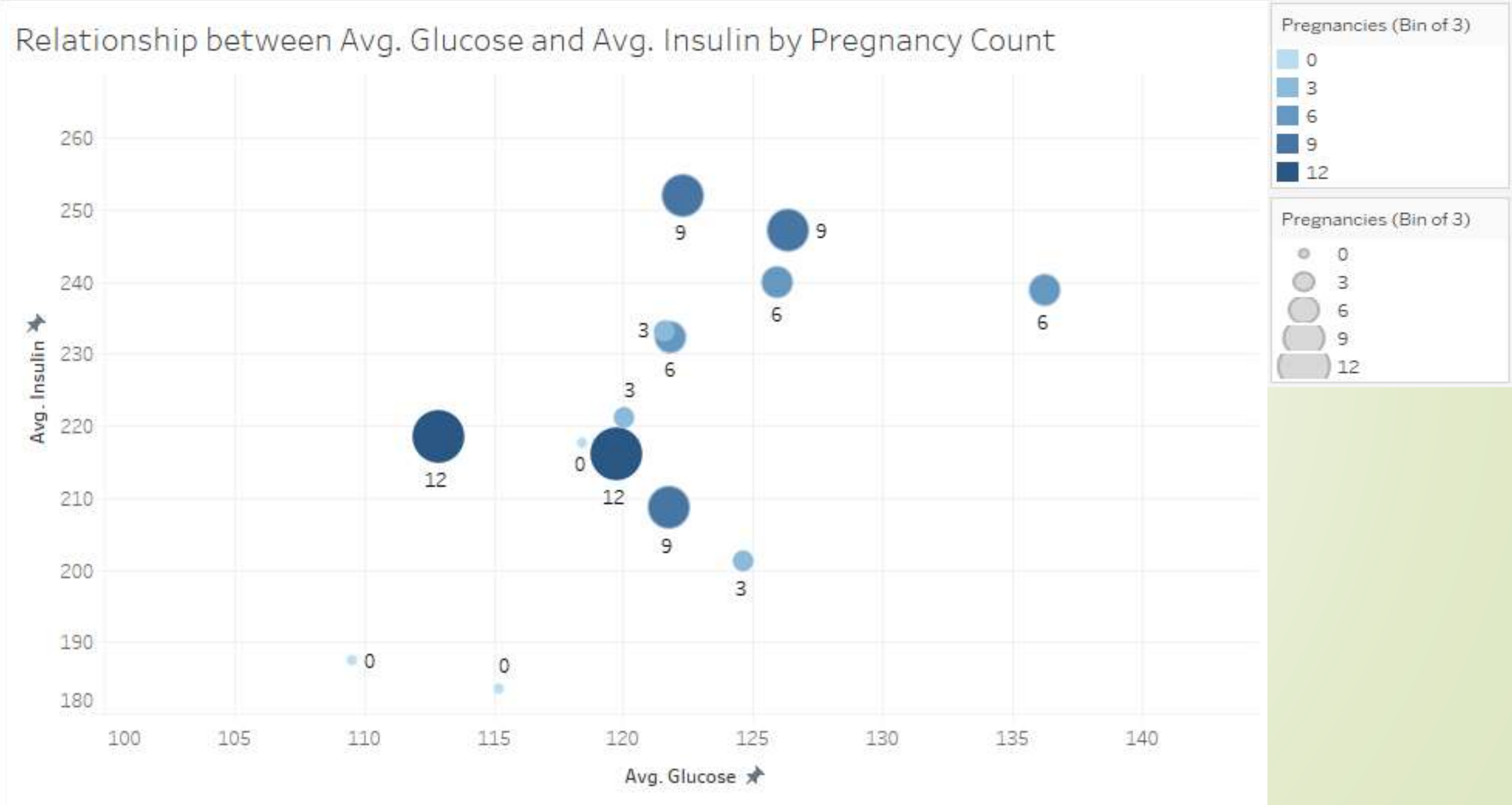


Tableau Report: Snips (Sheets)

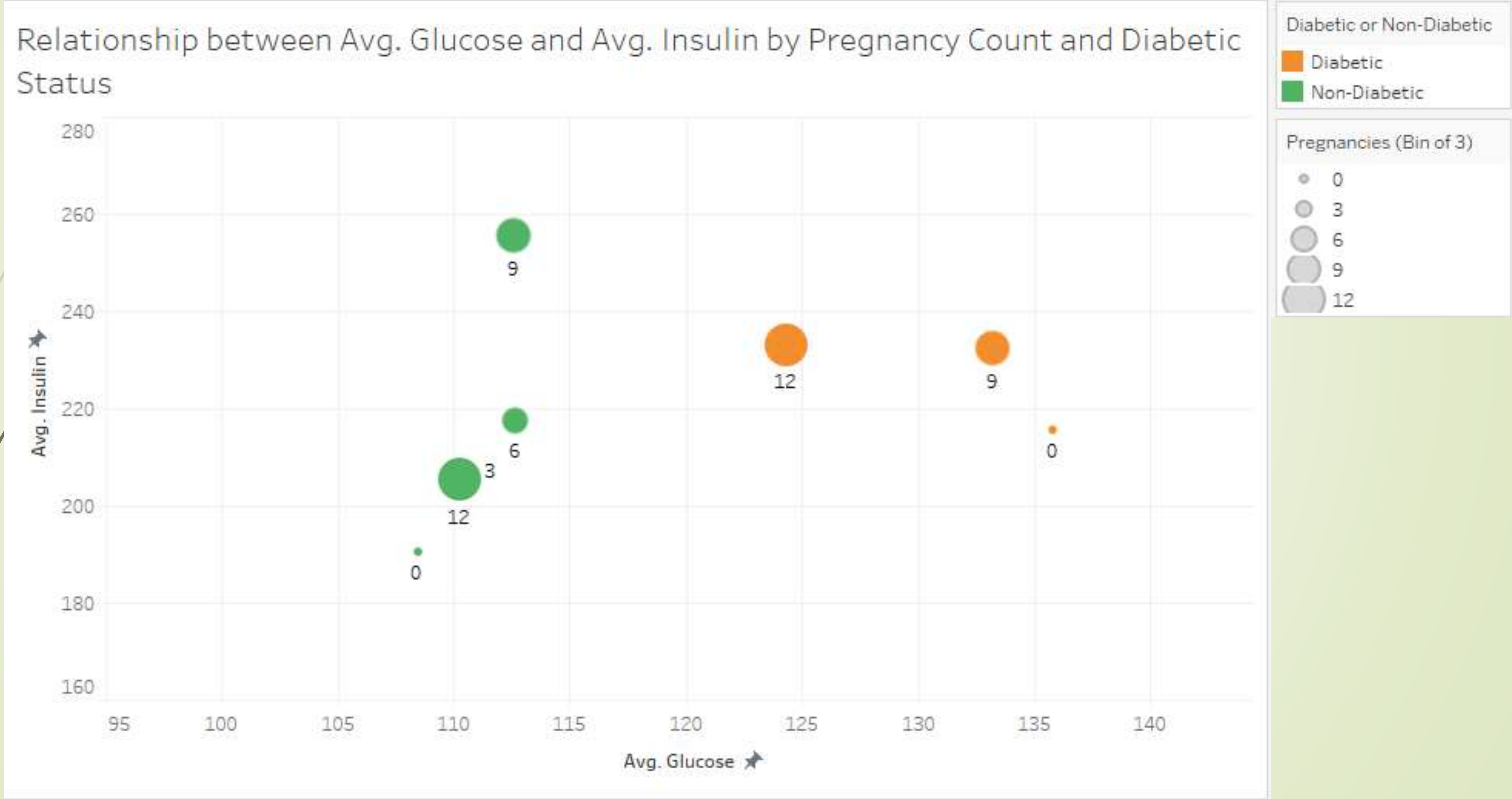
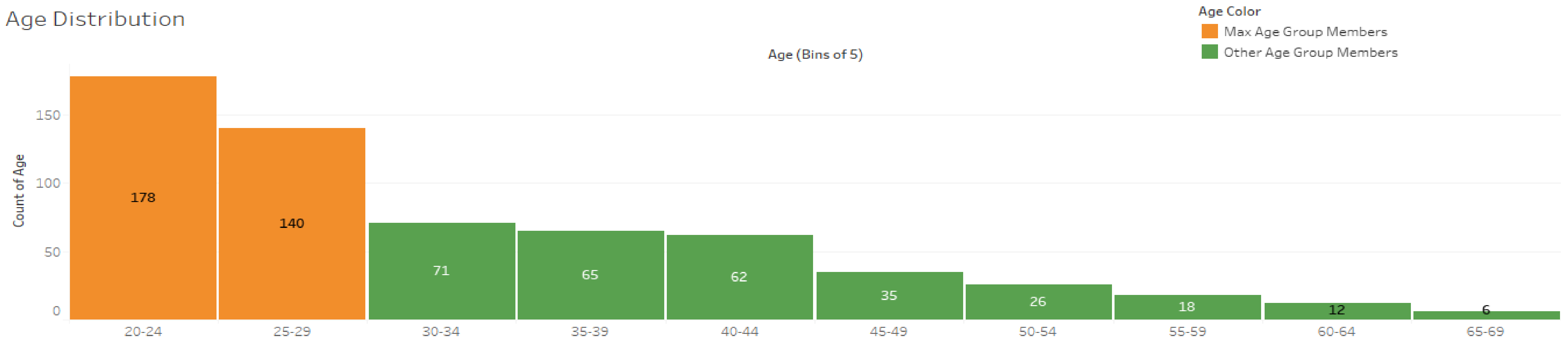


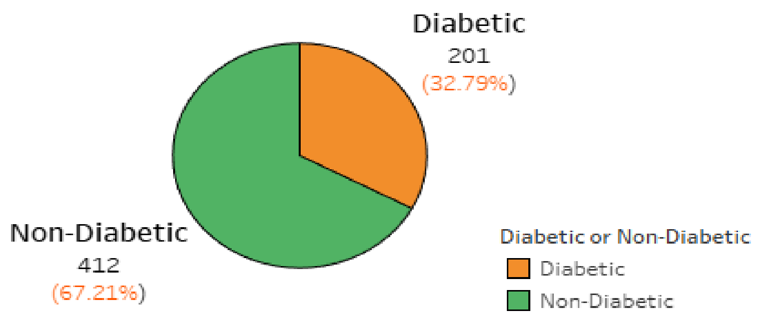
Tableau Report: Snips (Dashboards)

Basic Diabetes Data Information

Age Distribution



Proportion of Diabetic Population



Average Glucose level based on Age

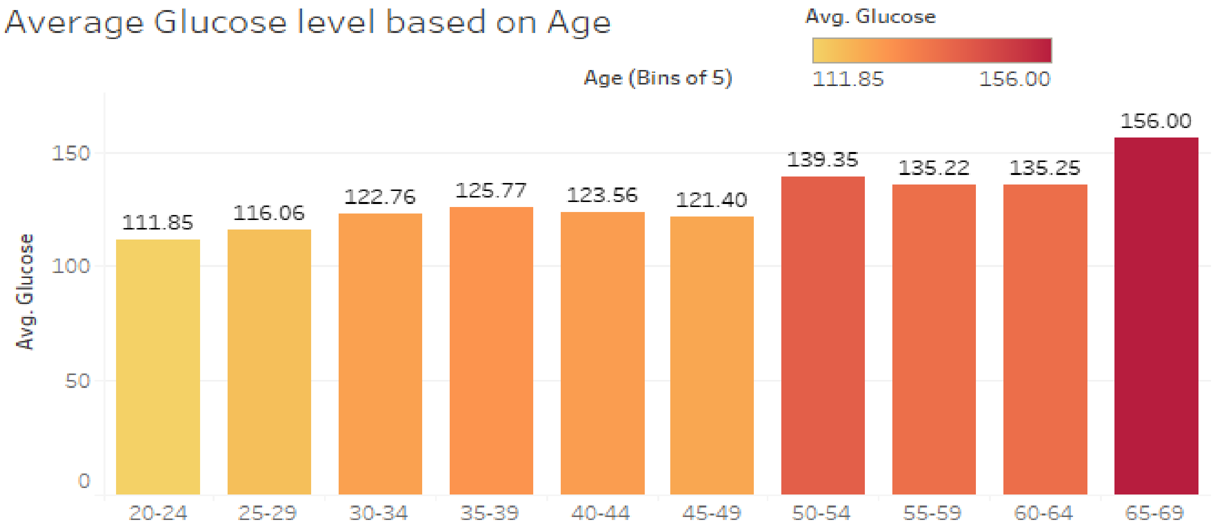


Tableau Report: Snips (Dashboards)

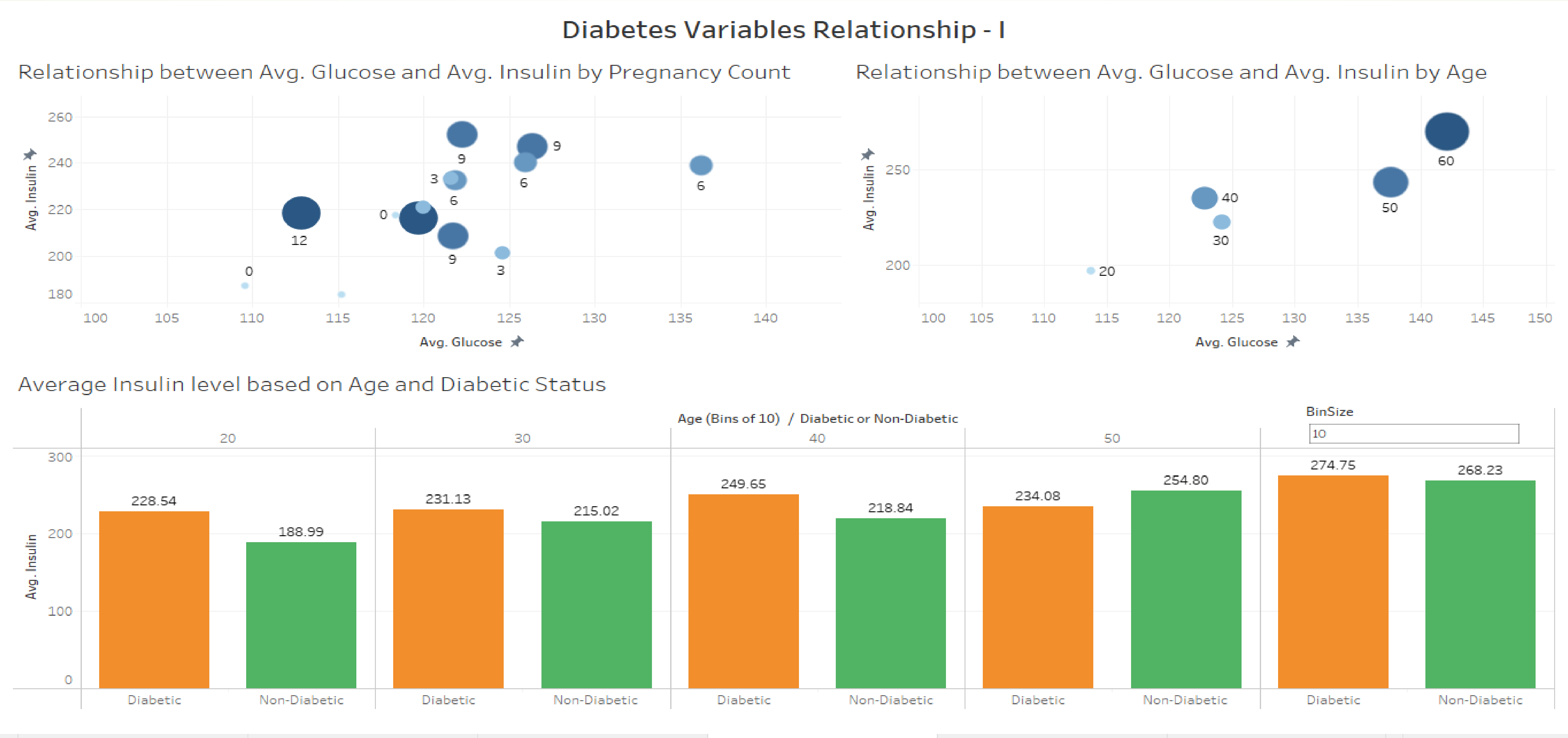


Tableau Report: Snips (Dashboards)

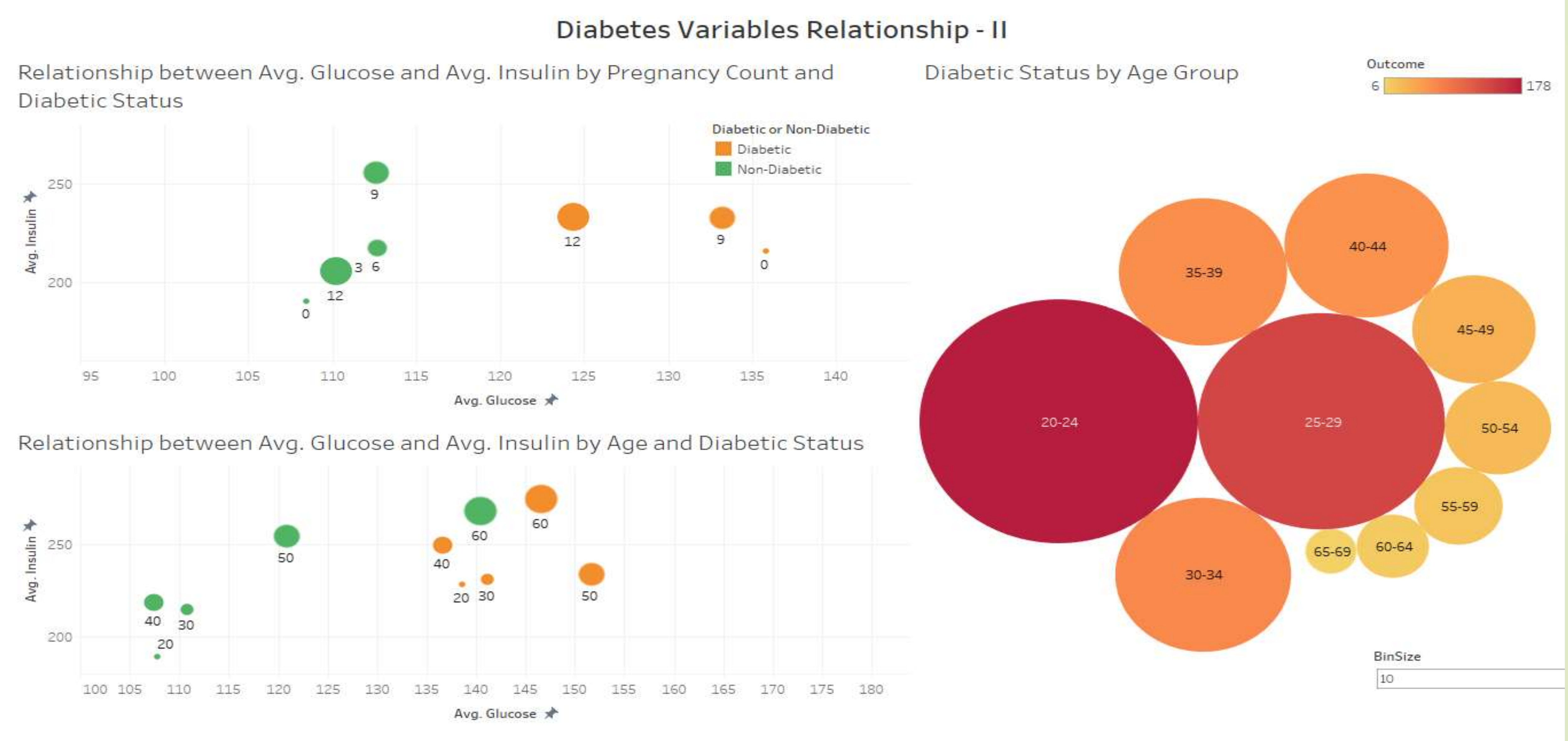


Tableau Report: Snips (Dashboards)

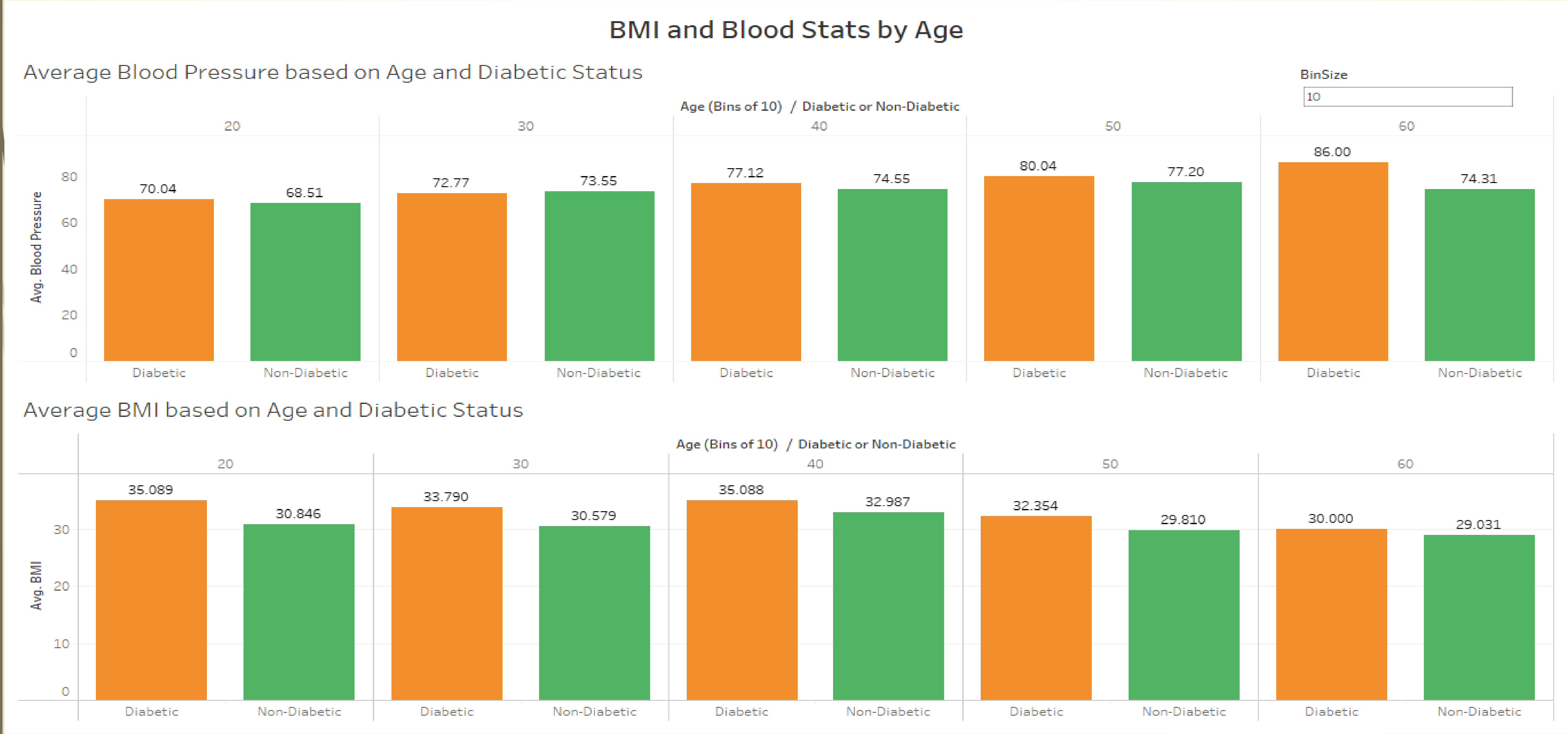


Tableau Report: Snips (Dashboards)

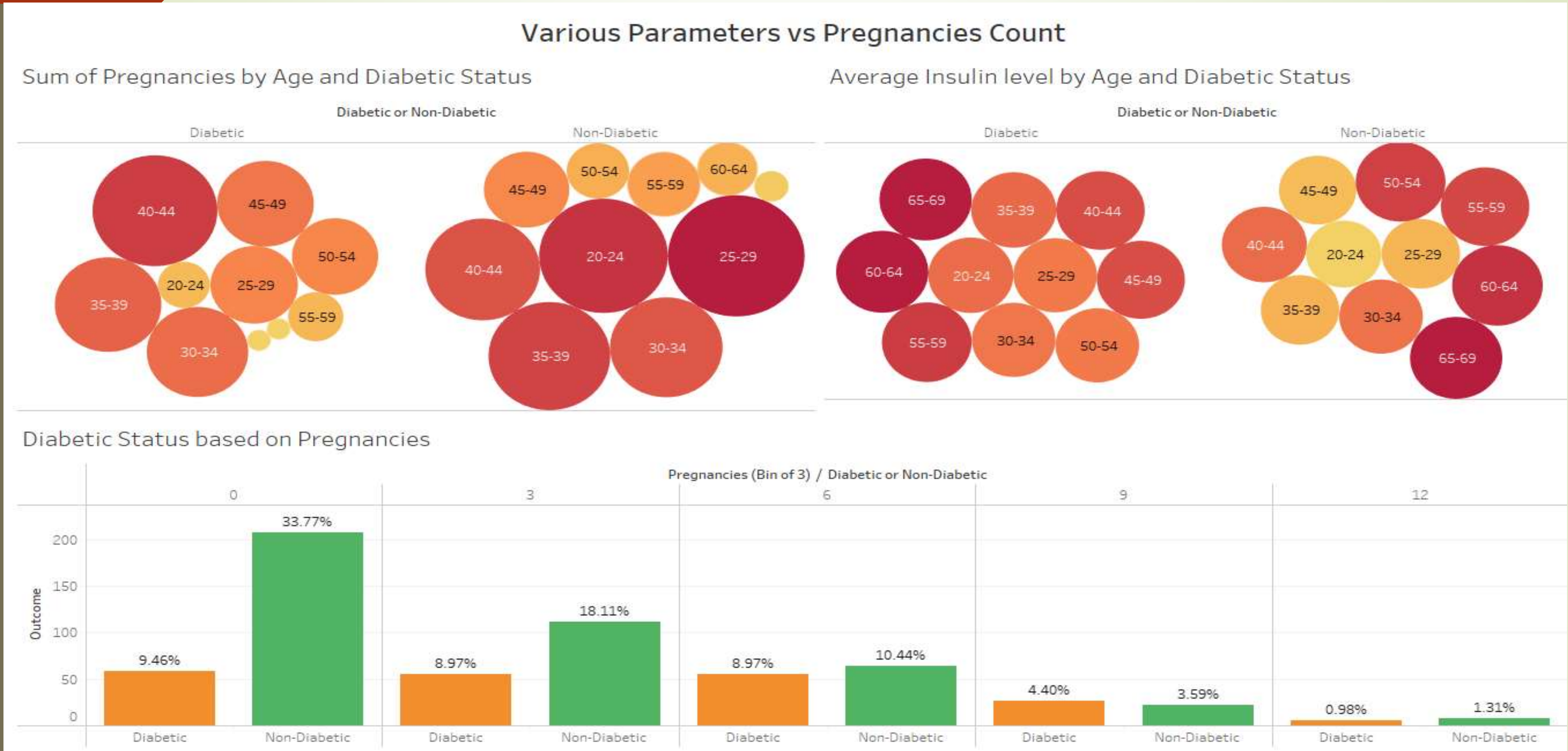


Tableau Report: Snips (Dashboards)

Various Parameters vs Age

Different Parameters based on Age

	Age (Bins of 5) 20-24	25-29	30-34	35-39	40-44	45-49	50-54	55-59	60-64	65-69
Avg. BMI	31.0	32.5	31.2	32.8	34.5	33.3	32.6	29.2	29.2	29.5
Avg. Blood Pressure	68.0	69.9	71.8	74.7	73.9	79.4	80.6	76.1	77.0	78.7
Avg. Diabetes Pedigree Function	0.4	0.4	0.5	0.4	0.4	0.4	0.5	0.5	0.4	0.4
Avg. Glucose	111.9	116.1	122.8	125.8	123.6	121.4	139.3	135.2	135.3	156.0
Avg. Insulin	190.5	204.4	228.3	215.4	242.2	222.3	235.3	255.4	267.7	274.7
Avg. Skin Thickness	27.0	29.1	28.8	30.6	30.3	30.4	28.5	28.6	27.3	29.1

Avg Insulin & Avg Glucose Comparison between Age Group and Diabetic Status

Tableau Report: Summary

- There are **32.79%** (201 observations) Diabetic population, where as **67.21%** (412 observations) population is Non-Diabetic.
- Maximum population is between **20-29 years** of age group where as very few people are **60+**.
- It is observed that higher glucose level was found on older age groups, and also glucose level increases with age.
- It is observed from the data that **higher** the pregnancies, lower is the population having **no** diabetes. However, in terms of diabetes, there is no such pattern spotted. Diabetes population is similar across 0-6 times pregnancy count, and it decreased with further bins.
- It was observed that for diabetic population, the **avg blood pressure** is **higher** for **older people**, and is relatively **consistent** on **non-diabetic** population.

Tableau Report: Summary

- Avg. **BMI** seems to slide **downwards** when plotted against age-bins for **diabetic** population, not observable though. However, it was **consistent** for **no-diabetic** population
- It was reported from the charts that average **Insulin** levels are **higher** for **diabetic** population, compared to **non-diabetic** population of same age-group
- **Majority** of the **diabetic** population is from **20-29** age group, however, this does not conclude that younger age group is prone to diabetes. More data is required to validate this fact.
- It was also observed that for diabetic population, the avg. **glucose** level is similar for all age groups, but shows trend in n**on-diabetic** population, and **increases** with age group. It may indicate <u>that lower levels of glucose may contribute to diabetes symptoms</u>.

Tableau Report: Summary

- It was also noticed that **significantly higher** levels of **insulin** and **glucose** may be expected from a **diabetic** patient. Which implies that if glucose and insulin are produced by the body heavily, it may be a case of diabetes.
- It was also observed that **higher pregnancies** has **higher** levels of average **insulin** and **glucose**, for diabetic population. (however, few outlier cases was also observed with 0 pregnancies.)
- Glucose is in general observed to be a significant factor in determining diabetes, therefore, **checking** the **glucose level** (along with **insulin level**) may help to **control diabetes**.

Diabetes
aged Tableau Work

Appendix

- Please refer 'PGP DS - Capstone Project - Healthcare - Diabetes' file, submitted along with this PPT
- Because the code was developed in jupyter notebook, it has source code along with the detailed analysis and report
- All the graphs included in this presentation can also be found in that project report
- This PPT is just a glimpse of the analysis done, for quick reference. Detailed work is present in the project report – "PGP DS - Capstone Project - Healthcare - Diabetes".
- Predicted values for 'test dataset' is attached as an csv file in slide number 63 of this PPT.
- Tableau Link is present in slide number 66 along with the dataset used for tableau report
- Tableau workbook is present in slide number 91

Thank you!