|  |  |
| --- | --- |
|  | Project 2 |
|  |  |
|  | Priyanka Nayak  Stat 6250: SAS Programming (Fall 2017) Prof. Eric A Suess  12/8/17 |

# Introduction

This project describes the in-depth analysis of data manipulation technique for MaxisIT

company for the job description below. MaxisIT is the only Integrated Software Platform for biopharmaceutical industry. They offer self-service data preparation and analytics products including clinical data repository, data management, data integration, statistical computing, advance analytics, reporting and visualization that are regulatory compliant, validated, cloud-based and delivered through alternate models viz. Enterprise SaaS, On-premise deployment, or as a Hybrid software-enabled service

Below is the link to job description

<https://www.glassdoor.com/job-listing/sr-sas-programmer-maxisit-JV_IC1147390_KO0,17_KE18,25.htm?jl=2306069588>

To study in depth, I will be using the below data manipulation technique or Proc in SAS

* Reading data values separated by Commas (CSV Files)
* SAS Graph(Cluster bar graph and Stacked bar graph)
* PROC SQL

I will be using **Diabetic dataset** to demonstrate data manipulation technique using SAS

## Sources for Dataset

## [https://archive.ics.uci.edu/ml/datasets/diabetes+130-us+hospitals+for+years+1999-2008](https://archive.ics.uci.edu/ml/datasets/diabetes+130-us+hospitals+for+years+1999-2008%20)

# Data Set Information

# The dataset represents 10 years (1999-2008) of clinical care at 130 US hospitals and integrated delivery networks. It includes over 50 features representing patient and hospital outcomes. Information was extracted from the database for encounters that satisfied the following criteria. (1) It is an inpatient encounter (a hospital admission). (2) It is a diabetic encounter, that is, one during which any kind of diabetes was entered to the system as a diagnosis. (3) The length of stay was at least 1 day and at most 14 days. (4) Laboratory tests were performed during the encounter. (5) Medications were administered during the encounter. The data contains such attributes as patient number, race, gender, age, admission type, time in hospital, medical specialty of admitting physician, number of lab test performed, HbA1c test result, diagnosis, number of medication, diabetic medications, number of outpatient, inpatient, and emergency visits in the year before the hospitalization, etc.

## Data Analysis

Using import procedure to read an external Excel file, the out= function creates an output data set in SAS temporary (work) library named diabetes.

**PROC** **IMPORT** OUT= Work.diabetes

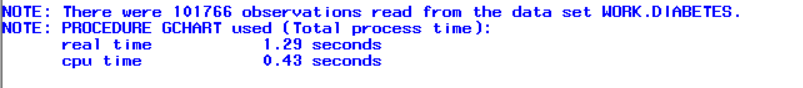
DATAFILE="C:\Users\Priyanka\Desktop\csu\SAS\datasets\diabeteic.xlsx"

DBMS=xlsx REPLACE;

GETNAMES=YES;

DATAROW=**2**;

**RUN;**



# **(Stat 6250) Write a SAS program to implement a data manipulation using Proc SQL.**

This project describes how to implement SAS Graphs using Clustered bar graph and Stacked bar graph using diabetes dataset.

SAS/GRAPH software is a very powerful tool for creating a wide range of business and scientific graphs. The types of graphs that can be produced with SAS/GRAPH software and the basic procedure syntax for creating some of them, including bar charts and plots we have to understand how to identify the appropriate variables of your data for the graph you want and how to customize its appearance.

## Cluster Bar Graph Program

\*/Program to study the diabetetic data set and implement using Cluster bar graph;

title "Average Diabetic patients by Gender and race using Cluster Bar Graph";

pattern1 value=L1;

**proc** **gchart** data=work.diabetes;

vbar Gender /

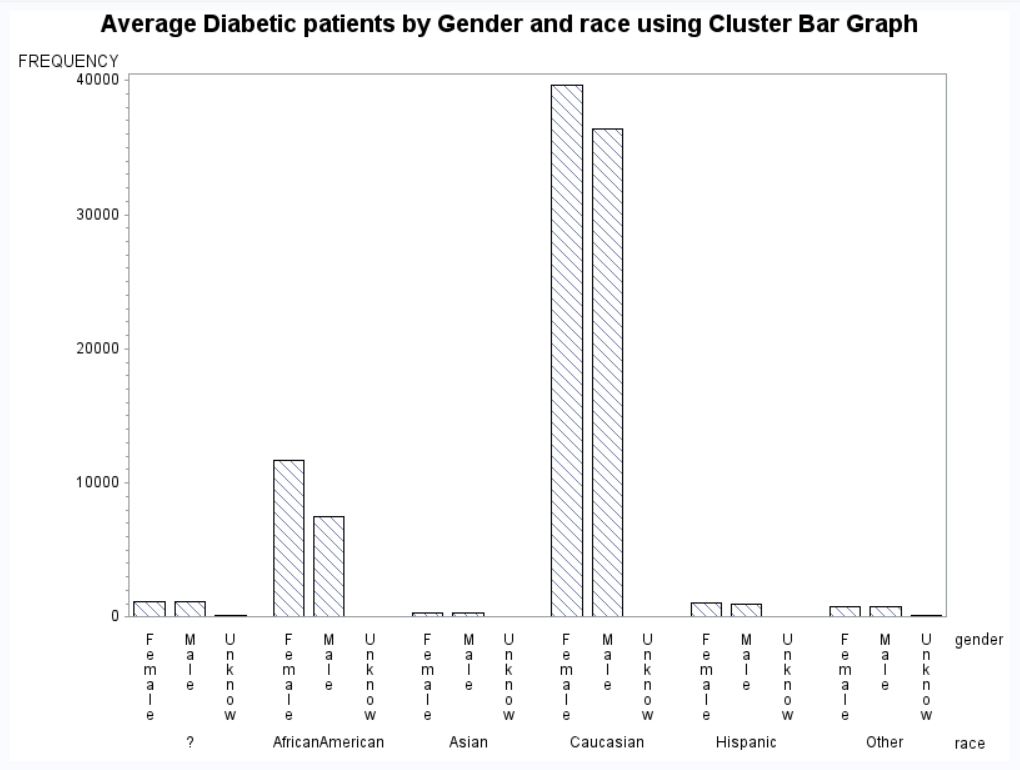
type=mean

group=race;

**run**;

**quit**;

### **Output:**



.

## **Stacked Bar Graph Program**

\*/Program to study the diabetic data set and implement using Stacked bar graph;

title "Average Diabetic patients by Gender and race using Stacked Bar Graph";

pattern1 value=L1;

pattern2 value=R3;

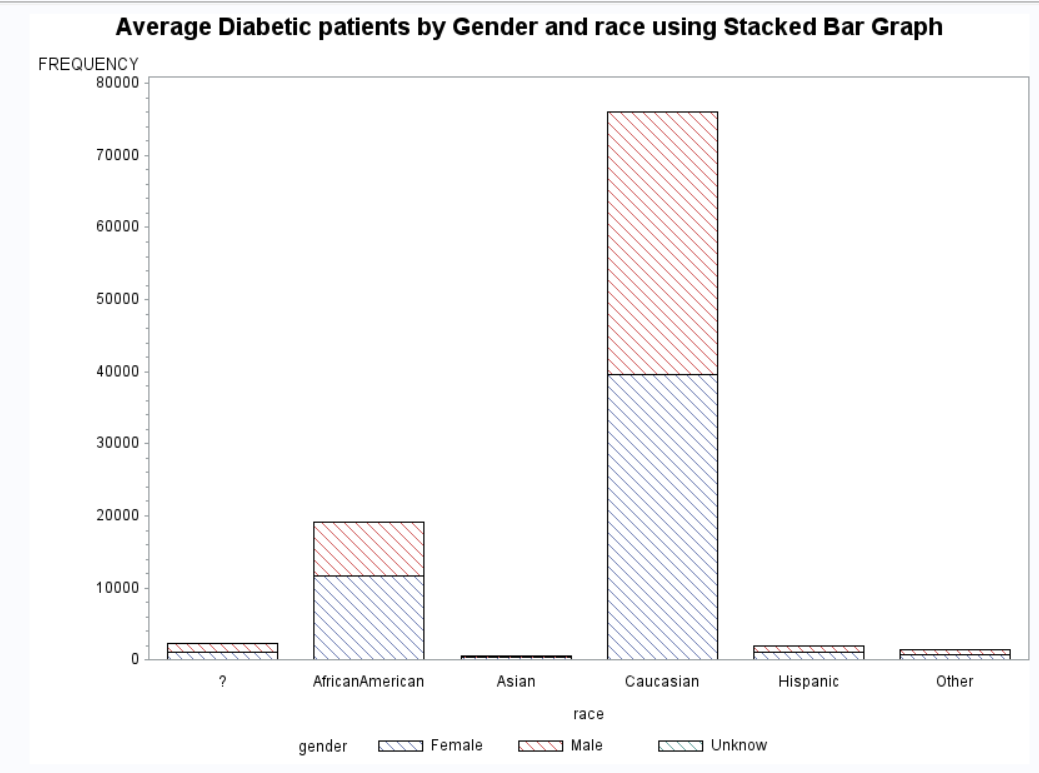
**proc** **gchart** data=work.diabetes;

vbar race / subgroup=Gender;

**run**;

**quit**;

### **Output:**

****

**PROC SQL:**

PROC SQL (structured query language) offers an alternative

to the DATA step for querying and combining SAS data sets. There are some tasks that

PROC SQL can perform much better and easier than the DATA step. Other tasks may be

easier or more efficient using a DATA step.

Step Overview:

Using the proc sql list the race and admission\_type\_id using diabetic dataset.

\*/Program to read the dataset and implement it using sql;

**proc** **sql**;

title "Listing of race in Order depending on admission type";

select race,admission\_type\_id

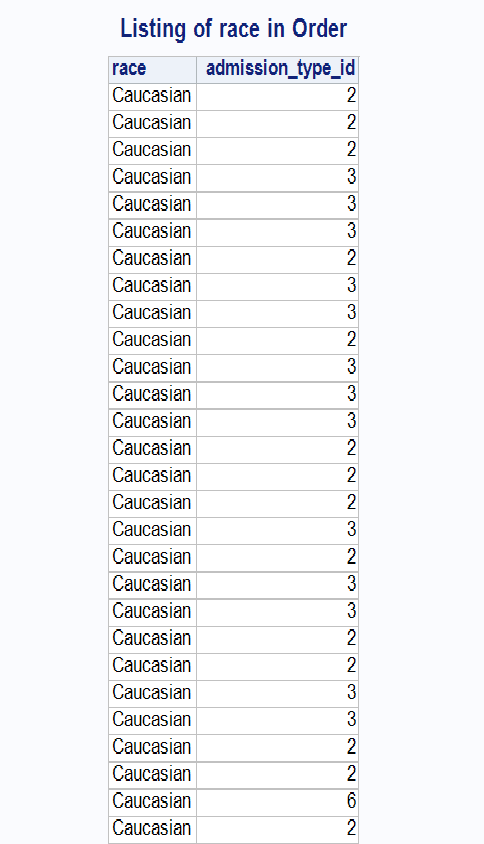
from work.diabeteic

where admission\_type\_id gt **1**

having race in ('Caucasian')

order by race;

**Output:**

****

**(Extra-credit) Write a SAS macro to be added to the class macro library.**

**%Let Macro function:**

Create macro variables by using the %LET statement, the SYMPUT routine, or the INTO clause in the SELECT statement from the SQL procedure. Sometimes a SAS programmer is often unsure when best to use which method due to a lack in understanding each step of macro language processing.

Step Overview: Create Macro variable %Let function to assign macro variable And used to display the diabetic analysis depending upon the race and gender.

\*/Program to read diabetic data set and implement it using macros;

%let var\_list = race gender;

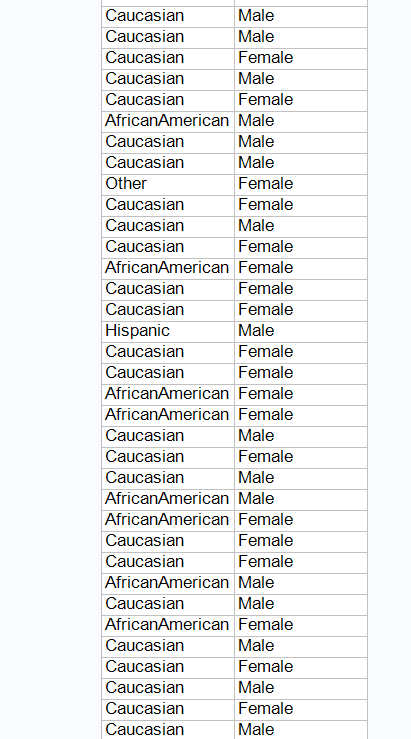
title "Using a Macro Variable List";

**proc** **print** data=work.diabeteic noobs;

var &var\_list;

**run**;

**Output:**

****

## Conclusion

The analysis of diabetes data set helped me to understand how the data manipulation techniques are used in analyzing the patient’s diabetes information using PROC GRAPHS and PROC SQL**.** Using **PROC** SQL which can perform much better and easier than the DATA step. PROC PLOT will give a graphical representation of data which helps us to study the dataset better.