# Final Project

*Umais Siddiqui*

*December 8, 2017*

## Final Project

Rpubs Link: http://rpubs.com/umais/data607-FinalProject

Github Link: https://github.com/umais/DATA-607/tree/master/FinalProject

### Overview

In this Project I will be analyzing a subset of dummy patients and associated pharmacy data retrieved from files in to a MYSQL Data store . All of this data is de identified data so nothing in this data identifies real patient data. One of my data sources will be csv file that I will be retrieving from a third party software called John Hopkins ACG System.

The purpose of the project is to use the Pharmacy data to determine the Health Risk of Patients and also determine what conditions the patients have based on the drugs being taken. Any company that deals with patients whether it is doctors, health insurance companies or Care Organizations can use this type of analysis to detemine which patients to focus their care coordination efforts on.

### Loading the Data in to MySQL from Patient.csv and Pharmacy.csv

I have set the eval value of the below r code to false as the below script is executed from a r Script file which is also a part of this final project. What this code is doing is a one time dump in to My Sql Tables Patient and Pharmacy from the CSV files which can also be found in the inputfiles directory.

```r
library(RMySQL)
library(lubridate)

killDbConnections <- function () {

  all_cons <- dbListConnections(MySQL())

  print(all_cons)

  for(con in all_cons)
    +  dbDisconnect(con)

  print(paste(length(all_cons), " connections killed."))

}

killDbConnections()

mydb = dbConnect(MySQL(), user='root', password='Welcome@1', dbname='finalproject', host='localhost')

Patient <- read.csv("inputFiles/patients.csv", header = TRUE,sep=",")

Patient$DOB<-as.Date(parse_date_time(Patient$DOB,"mdy"))
```

```
head(Patient)

dbWriteTable(mydb, "patient", Patient, overwrite=FALSE,append=TRUE,row.names=FALSE)
Pharmacy <- read.csv("inputFiles/pharmacy.csv", header = TRUE,sep=",")

Pharmacy$FillDate<-as.Date(parse_date_time(Pharmacy$FillDate,"mdy"))

head(Pharmacy)

dbWriteTable(mydb, "prescriptionfillhistory", Pharmacy, overwrite=FALSE,append=TRUE,row.names=FALSE)
```

### Selecting the Patient Data and Pharmacy Data loaded

In the Code below we will verify that the data loaded sucessfully and we can view the records in the tables.

```
mydb = dbConnect(MySQL(), user='root', password='Welcome@1', dbname='finalproject', host='localhost')
rs = dbSendQuery(mydb, "SELECT * FROM patient;")

PatientData=fetch(rs, n=-1)

head(PatientData)
```

```
##   BeneficiaryNumber FirstName  LastName        DOB
## 1       99350001797      PHIL    JERASA 1948-05-05
## 2       99350002197 CENTRELLA    DINANT 1951-01-04
## 3       99350002397         J     MERRY 1942-06-15
## 4       99350005697    DONITA FERREY JR 1939-09-29
## 5       99350005897      JADA  CHARNICK 1926-05-11
## 6       99350006297     AVRUM    HUMMON 1942-09-18
```

```
rs = dbSendQuery(mydb, "SELECT * FROM PrescriptionFillHistory;")

PrescriptionHistory=fetch(rs, n=-1)

head(PrescriptionHistory)
```

```
##   BeneficiaryNumber         NDC        NPI   FillDate Supply
## 1       99350001797   955100410 8951265221 2016-06-18      4
## 2       99350001797 16729018317 2448858131 2016-08-13     30
## 3       99350001797 65862056090 4131612881 2016-06-30     30
## 4       99350001797    93965201 2046894101 2016-09-11      5
## 5       99350001797    93220305 8951265221 2016-05-26      1
## 6       99350001797   406012301 8951265221 2016-06-14      3
```

### Loading the ACG data from ACG output File

John Hopkins ACG takes two files as input the patient File and the Pharmacy File. The Pharmacy file is optional but it is required for the purposes of this project. John Hopkins ACG Software processes these two files and then produces two output files one file is the ACG output file and the other one is ACG Conditions File. We will be now analyzing the output that was produced by ACG

```
PatientACG <- read.csv("inputFiles/ACGOutput.acgd.csv", header = TRUE,sep=",")
```

```
head(PatientACG$rescaled_pharmacy_cost_predicted_risk)
```

```
## [1] 0.6047990 1.7241220 0.6378245 0.5098452 0.4584118 3.0624195
```

In the data returned by John Hopkins ACG Software I will be looking at the column rescaled__pharmacy__cost__predicted__risk. I will calculate the mean and Standard Deviation for this column so that I can use that to place the patients in different categories whether they are very low risk, low risk, medium risk , High Risk or Very High Risk.

What I will do here is use my own algorithm to calculate the Risk Level for a patient. As Per John Hopkins ACG software the Mean of the Rescaled Pharmacy cost is 1. So if the value is 1 in this column that means this is how much an average patient will cost based on the drugs they are taking.

The algorithm is below

if a patient cost is 0.5 then I am categorizing them as very low (1).

If the patient is greater than 0.5 but less than equal to the mean 1 then they are low risk (2).

If the patient cost is the mean + standard deviation then they are Medium Risk (3)

If the patient cost index is mean plus standard deviation multiply by 3 then they are High Risk

Finally if they are higher than that then they are very risky.

```
CalculateRiskLevel <- function(predictedCost,mean,sdvcost)
{
  if(predictedCost<0.5)
   return(1)
  if(predictedCost<=1)
    return(2)
  if(predictedCost<=mean+sdvcost)
    return(3)
  if(predictedCost<=(mean+(sdvcost*3)))
    return(4)
  else
    return(5)
}
Mean<-mean(PatientACG$rescaled_pharmacy_cost_predicted_risk)

SDVCost<-sd(PatientACG$rescaled_pharmacy_cost_predicted_risk)

Mean
```

```
## [1] 0.9110616
```

```
SDVCost
```

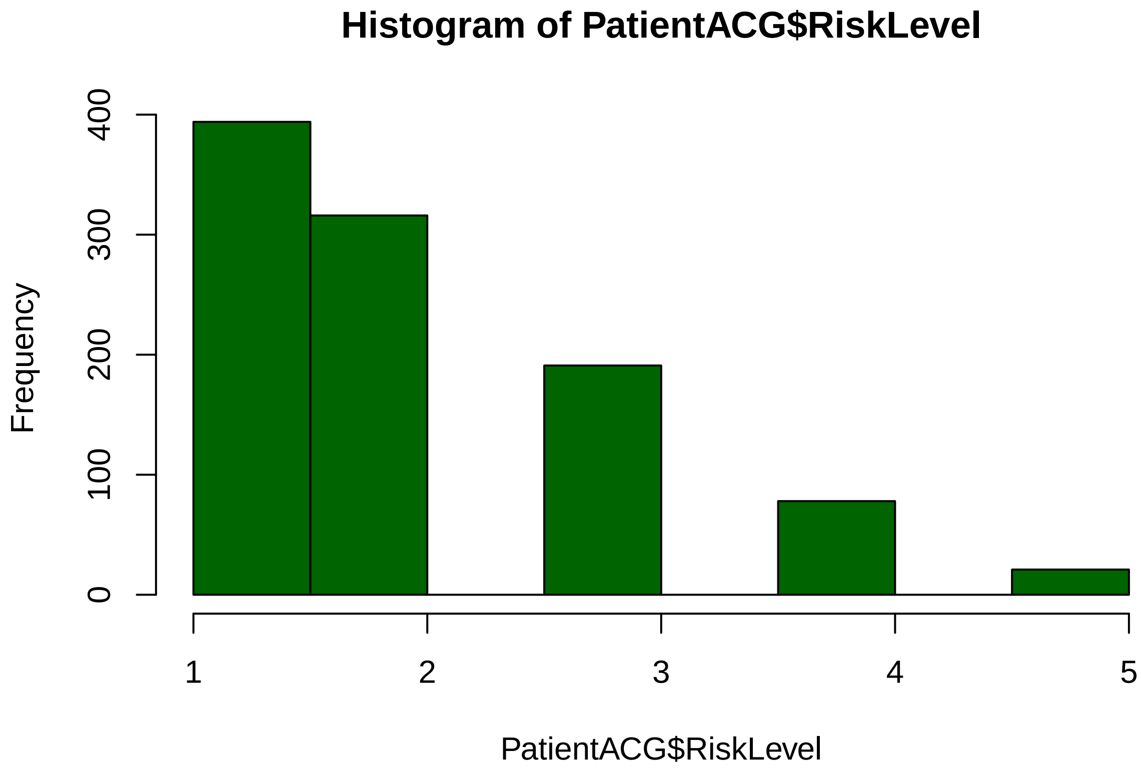
```
## [1] 0.9699812
```

```
PatientACG$RiskLevel= sapply(PatientACG$rescaled_pharmacy_cost_predicted_risk, CalculateRiskLevel,Mean,S

head(PatientACG$RiskLevel)
```

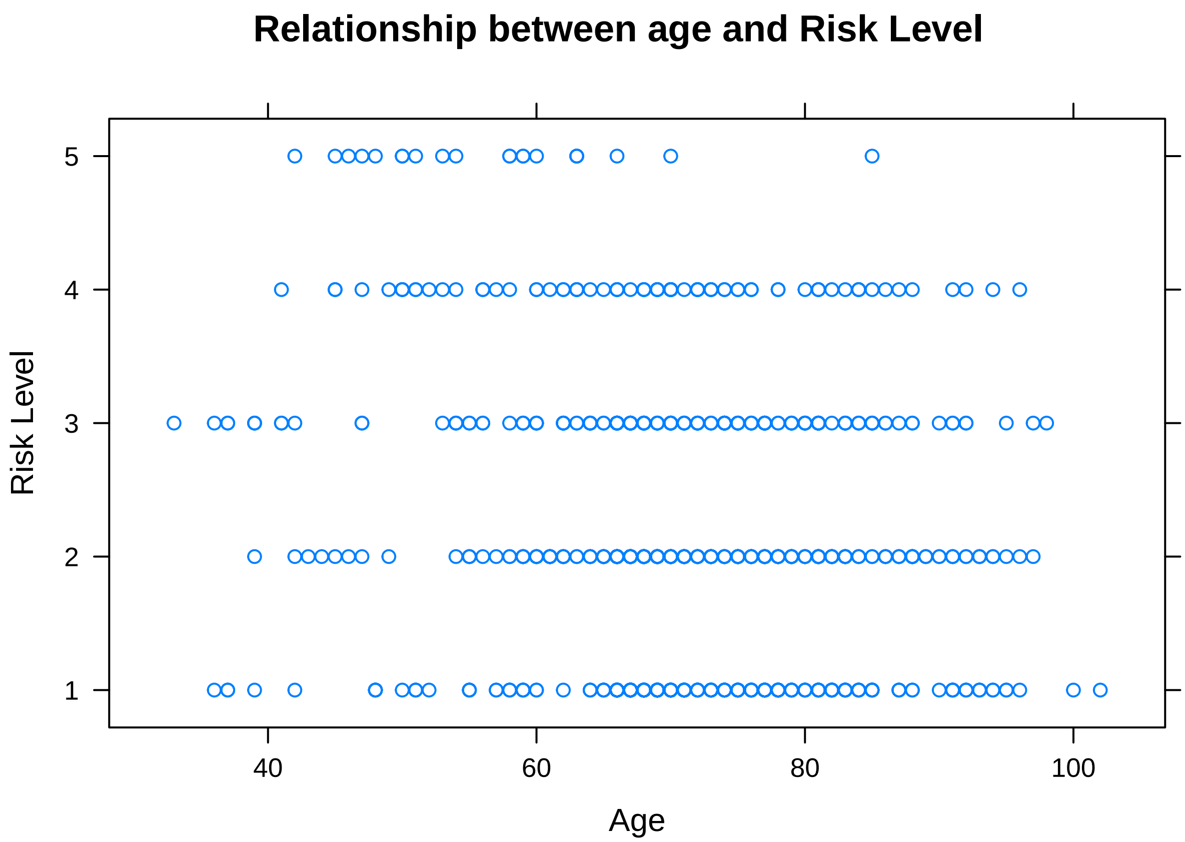
```
## [1] 2 3 2 2 1 4
```

### Histogram of the Risk Level

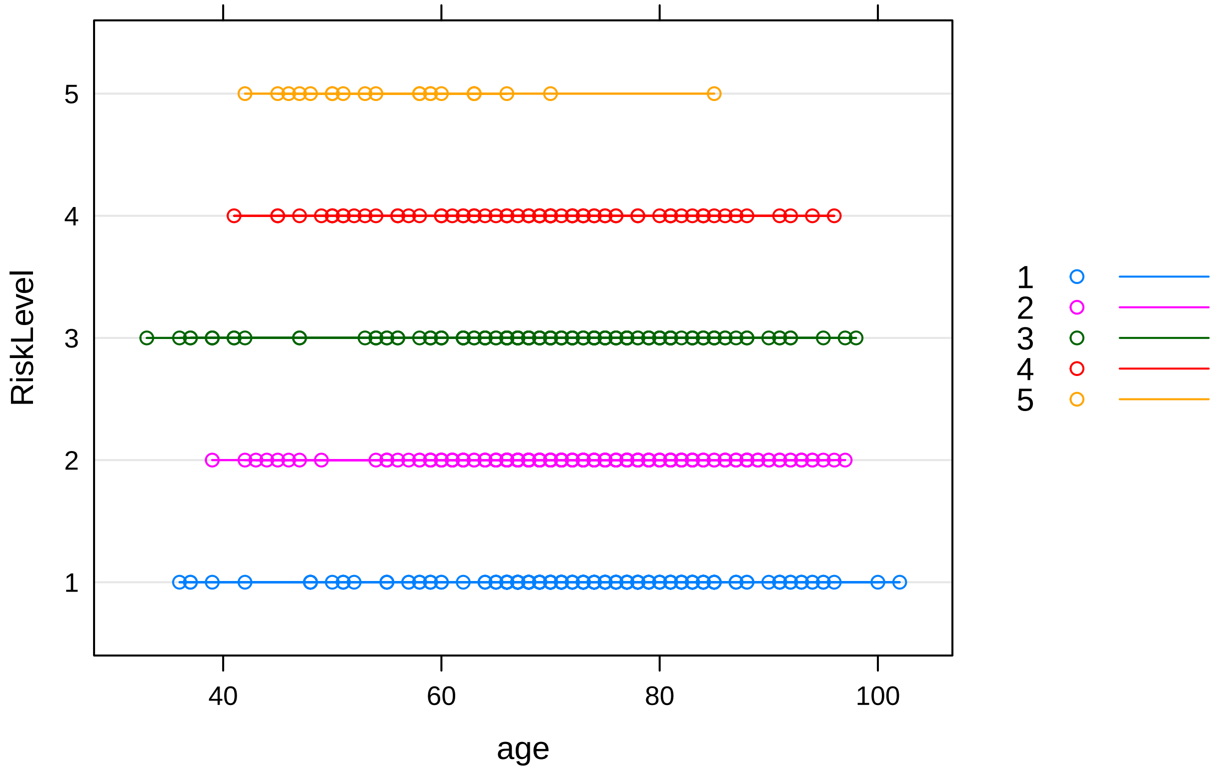
```
hist(PatientACG$RiskLevel, col="darkgreen")
```



```
xyplot(RiskLevel ~ age, data = PatientACG,
  xlab = "Age",
  ylab = "Risk Level",
  main = "Relationship between age and Risk Level")
```



```
dotplot(RiskLevel ~ age, data = PatientACG, groups = RiskLevel, type = "o",
auto.key = list(space = "right", points = TRUE, lines = TRUE))
```



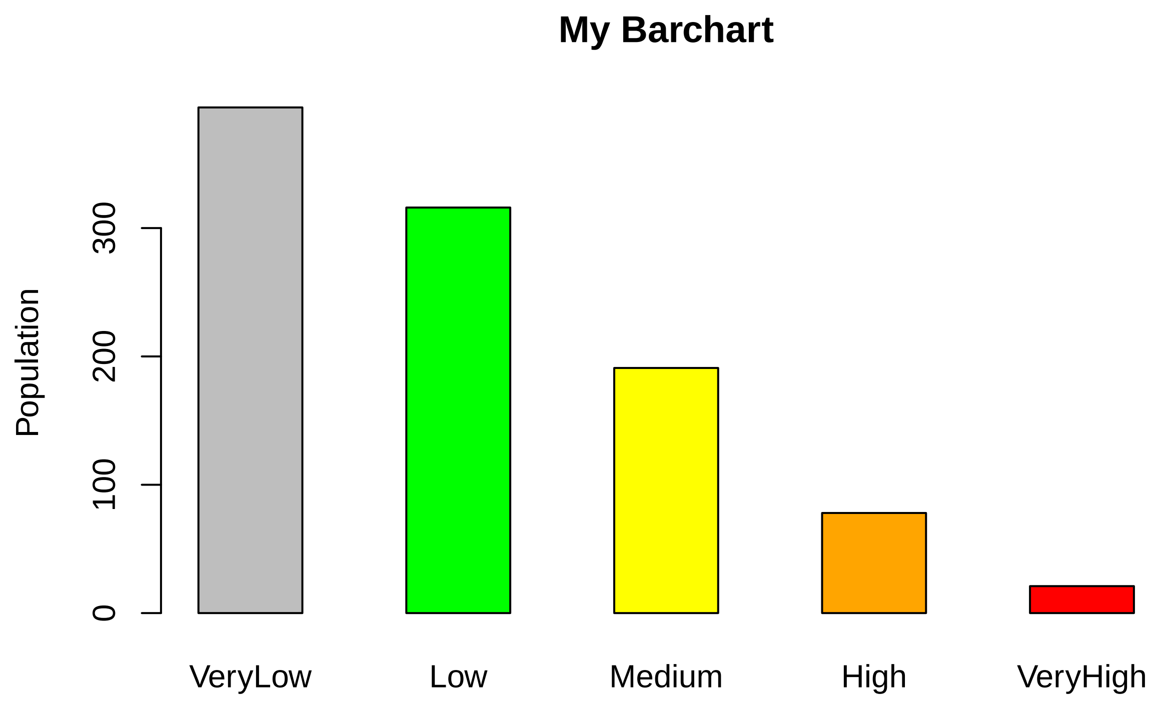
### Barchart Showing Counts of different Risk Levels

In the below code we will be creating counts of the population categorizing them in different buckets from Risk Level very low to Very High

```
colours <- c("grey", "green", "yellow", "orange", "red")

VeryLow = nrow(subset(PatientACG, RiskLevel==1))
Low = nrow(subset(PatientACG, RiskLevel==2))
Medium = nrow(subset(PatientACG, RiskLevel==3))
High = nrow(subset(PatientACG, RiskLevel==4))
VeryHigh = nrow(subset(PatientACG, RiskLevel==5))
t=data.frame(VeryLow,Low,Medium,High,VeryHigh)
t
```

```
##   VeryLow Low Medium High VeryHigh
## 1     394 316    191   78       21
```

```
barplot(as.matrix(t), main="My Barchart", ylab = "Population", beside=TRUE, col=colours)
```



We can see based on the above results that there are very few patients that are in the Very High category. This will narrow down the population that care coordination organizations have to focus on.

### Now Loading the ACG Conditions File based on NDC code from Pharmacy File.

```
PatientConditions <- read.csv("inputFiles/ACG_Conditions.csv", header = TRUE,sep=",")

head(PatientConditions)
```

```
##   patient_id rxmg_code                                 rxmg_description
## 1  9.935e+10   CARx030             Cardiovascular / High Blood Pressure
## 2  9.935e+10   CARx050              Cardiovascular / Vascular Disorders
## 3  9.935e+10   GASx060        Gastrointestinal/Hepatic / Peptic Disease
## 4  9.935e+10   GSIx010 General Signs and Symptoms / Nausea and Vomiting
## 5  9.935e+10   GSIx020                General Signs and Symptoms / Pain
## 6  9.935e+10   INFx020                         Infections / Acute Minor
##   major_rxmg_code     major_rxmg_description rxmg_impact_type
## 1             CAR             Cardiovascular                L
## 2             CAR             Cardiovascular                H
## 3             GAS   Gastrointestinal/Hepatic                M
## 4             GSI General Signs and Symptoms                M
## 5             GSI General Signs and Symptoms                M
## 6             INF                 Infections                L
##   rxmg_medical_source rxmg_pharmacy_source
## 1                   N                    Y
## 2                   N                    Y
## 3                   N                    Y
## 4                   N                    Y
## 5                   N                    Y
## 6                   N                    Y
```

As we can see that John Hopkins ACG software also produced result that can help tell us what conditions a patient has. Out of these conditions I would like to analyze the data for some conditions such as High Blood Pressure and Diabetes without Insulin.

The RxMGCode is a proprietry code from John Hopkins that is used to identify a co-morbidity (condition). In order to analyze the conditions I mentioned above I will be using the RxMG Code associated with those two conditions.

The two RxMG Codes of interest are CARx030 and ENDx040

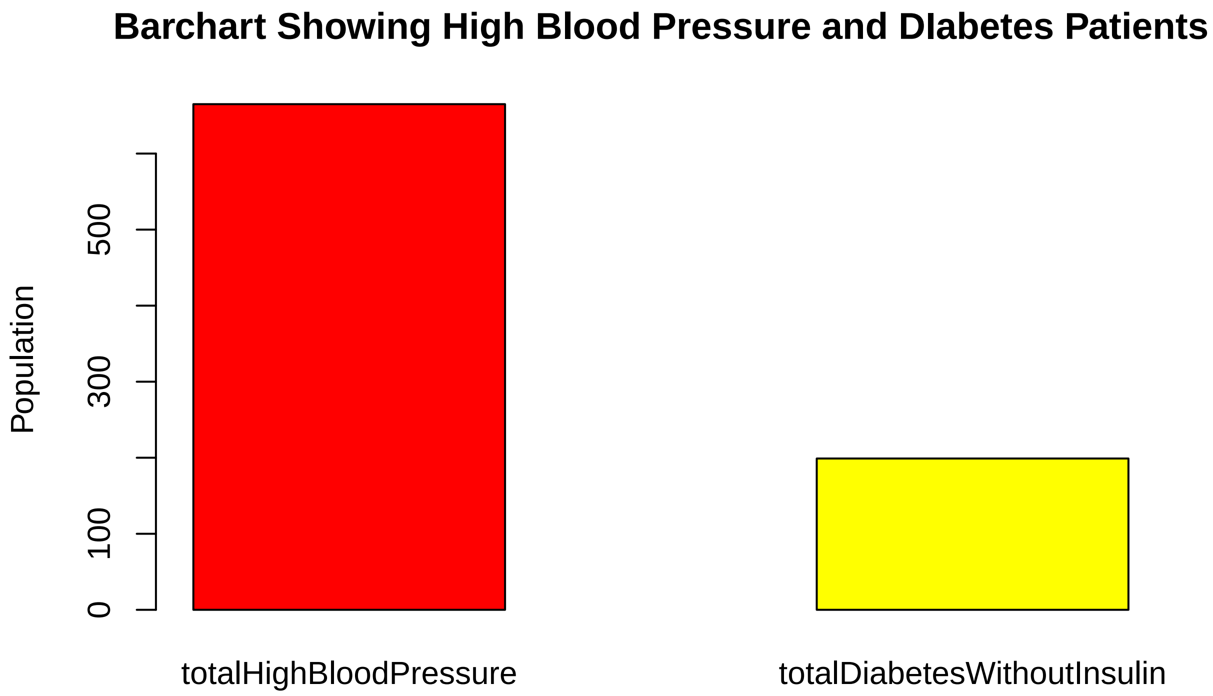
```
HighBloodPressure = subset(PatientConditions, rxmg_code=="CARx030")
DiabetesWithoutInsulin= subset(PatientConditions, rxmg_code=="ENDx040")

head(HighBloodPressure)
```

```
##     patient_id rxmg_code                     rxmg_description
## 1  99350001797   CARx030 Cardiovascular / High Blood Pressure
## 16 99350002397   CARx030 Cardiovascular / High Blood Pressure
## 19 99350002497   CARx030 Cardiovascular / High Blood Pressure
## 23 99350004197   CARx030 Cardiovascular / High Blood Pressure
## 28 99350004697   CARx030 Cardiovascular / High Blood Pressure
## 40 99350005697   CARx030 Cardiovascular / High Blood Pressure
##    major_rxmg_code major_rxmg_description rxmg_impact_type
## 1              CAR         Cardiovascular                L
## 16             CAR         Cardiovascular                L
## 19             CAR         Cardiovascular                L
## 23             CAR         Cardiovascular                L
## 28             CAR         Cardiovascular                L
## 40             CAR         Cardiovascular                L
##    rxmg_medical_source rxmg_pharmacy_source
## 1                    N                    Y
## 16                   N                    Y
## 19                   N                    Y
## 23                   N                    Y
## 28                   N                    Y
## 40                   N                    Y
```

```
head(DiabetesWithoutInsulin)
```

```
##     patient_id rxmg_code                     rxmg_description
## 9  99350002197   ENDx040 Endocrine / Diabetes Without Insulin
## 18 99350002397   ENDx040 Endocrine / Diabetes Without Insulin
## 53 99350006497   ENDx040 Endocrine / Diabetes Without Insulin
## 58 99350007097   ENDx040 Endocrine / Diabetes Without Insulin
## 66 99350007797   ENDx040 Endocrine / Diabetes Without Insulin
## 71 99350008397   ENDx040 Endocrine / Diabetes Without Insulin
##    major_rxmg_code major_rxmg_description rxmg_impact_type
## 9              END              Endocrine                L
## 18             END              Endocrine                L
## 53             END              Endocrine                L
## 58             END              Endocrine                L
## 66             END              Endocrine                L
## 71             END              Endocrine                L
##    rxmg_medical_source rxmg_pharmacy_source
## 9                    N                    Y
## 18                   N                    Y
## 53                   N                    Y
## 58                   N                    Y
## 66                   N                    Y
## 71                   N                    Y
```

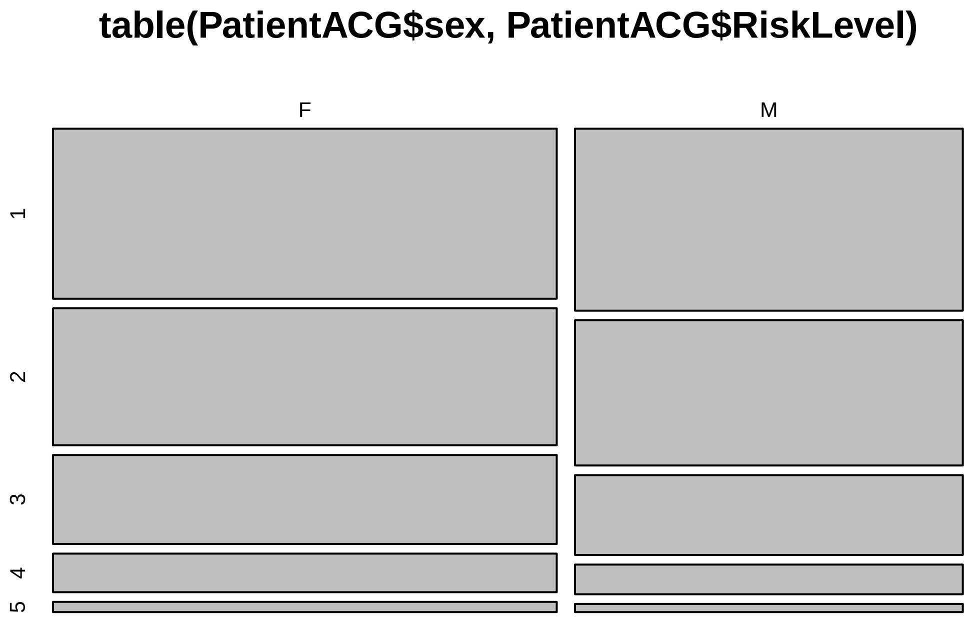
```
colours <- c("red","yellow")
totalHighBloodPressure=nrow(HighBloodPressure)

totalDiabetesWithoutInsulin=nrow(DiabetesWithoutInsulin)

data=data.frame(totalHighBloodPressure,totalDiabetesWithoutInsulin)
barplot(as.matrix(data), main="Barchart Showing High Blood Pressure and DIabetes Patients", ylab = "Popu
```



### Doing some more analysis

I want to do some further analysis based on the gender of the patient the risk level

```
mosaicplot(table(PatientACG$sex,PatientACG$RiskLevel))
```



We can see the distribution of the Female and male in different Risk levels

One more thing I want to check is what percentage of female out of the population are at high risk versus the male.

```
TotalPatients=nrow(PatientACG)

totalFemaleHighRisk=nrow(subset(PatientACG,sex=="F" & RiskLevel==5))
totalMaleHighRisk=nrow(subset(PatientACG,sex=="M" & RiskLevel==5))
totalFemaleHighRisk
```

```
## [1] 13
```

```
totalMaleHighRisk
```

```
## [1] 8
```

```
print(paste("Percentage of Male with High Risk" , (totalMaleHighRisk/TotalPatients)*100.0))
```

```
## [1] "Percentage of Male with High Risk 0.8"
```

```
print(paste("Percentage of Female with High Risk" , (totalFemaleHighRisk/TotalPatients)*100))
```

```
## [1] "Percentage of Female with High Risk 1.3"
```

### Conclusion

We can see based on the results that we can use the pharmacy data to determine the Risk Level of the Patients as well as see what Implied conditions they may have based on the drugs being taken. This is a very helpful way to get an insight in to what the patients condition is.

We can take this a step further and try to determine the gaps in prescription fills of the patients and see if they are being adherent in filling there prescriptions. We will not be doing that in this project but I think it is something worth looking in to for future Projects.