RankMiner: Predicting Phone Agent Attrition

**Group 2**

Mohammed Ihtesham

Deepen Panchal

Vishal Punjabi

Paridhi Varma

**Draft Update:** 04/05/2016

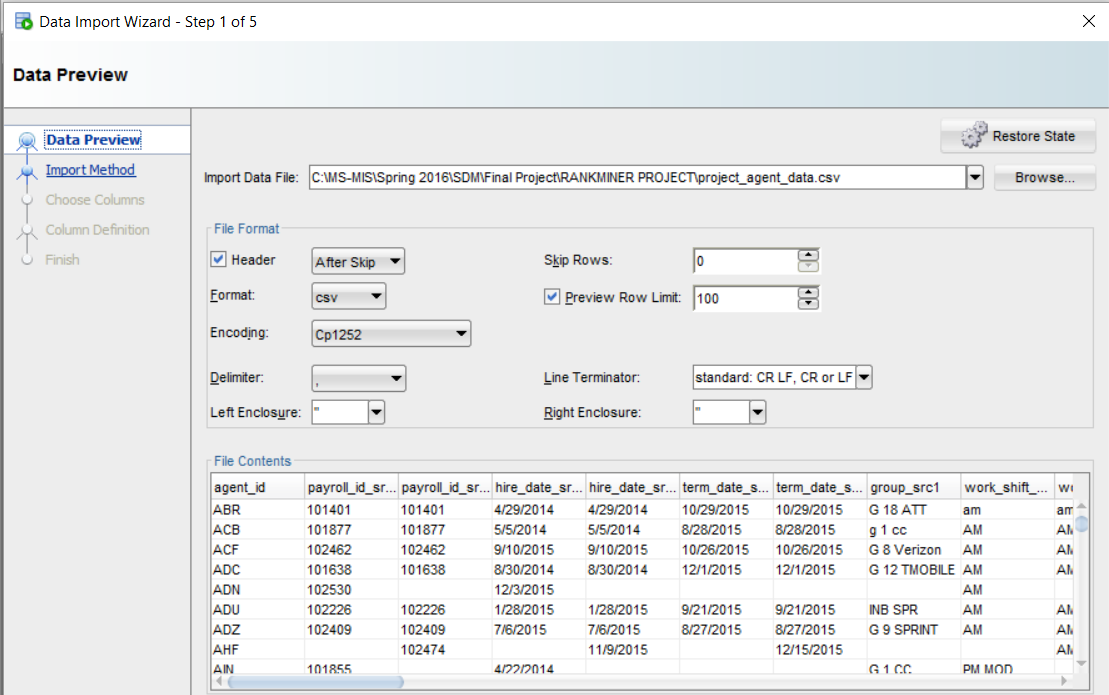
**Summary of Activities**

The group met on 3rd April (Sunday) and 4th April (Monday) to plan and discuss the project. All group members were present at both the meetings. Following is the summary of activities performed as of 5th April 2016.

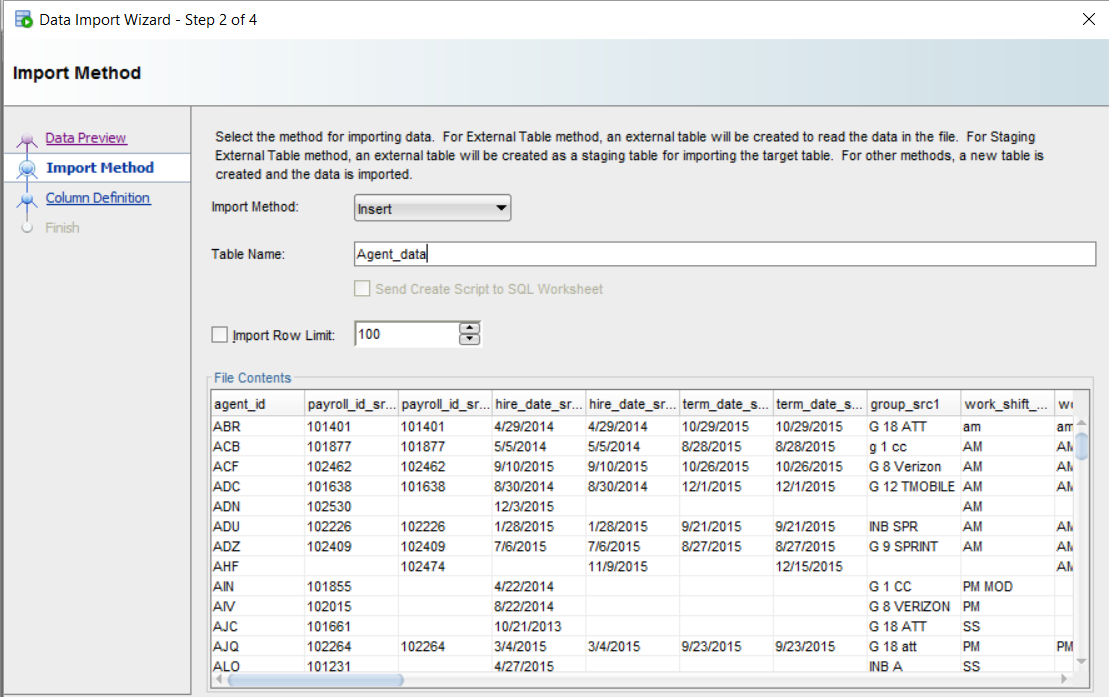
|  |  |  |  |
| --- | --- | --- | --- |
| **Date** | **Task** | **No. of Hrs.** | **Status** |
| 04-07-2016 | Load all data files in Oracle Relational database | 2 | Completed |
| 04-08-2016 | Merge Data of source 1 and source 2 | 1.5 | Completed |
| 04-09-2016 | Fill in the gaps by inserting Null and Unknown values | 1.5 | Completed |
| 04 -11-2016 | Data cleaning activity similarly for Call data and Feature Data | 1 | Completed |
| 04-12-2016 | Develop visualizations for cleaned data. | 2 | In progress |

Load data files in Oracle Database

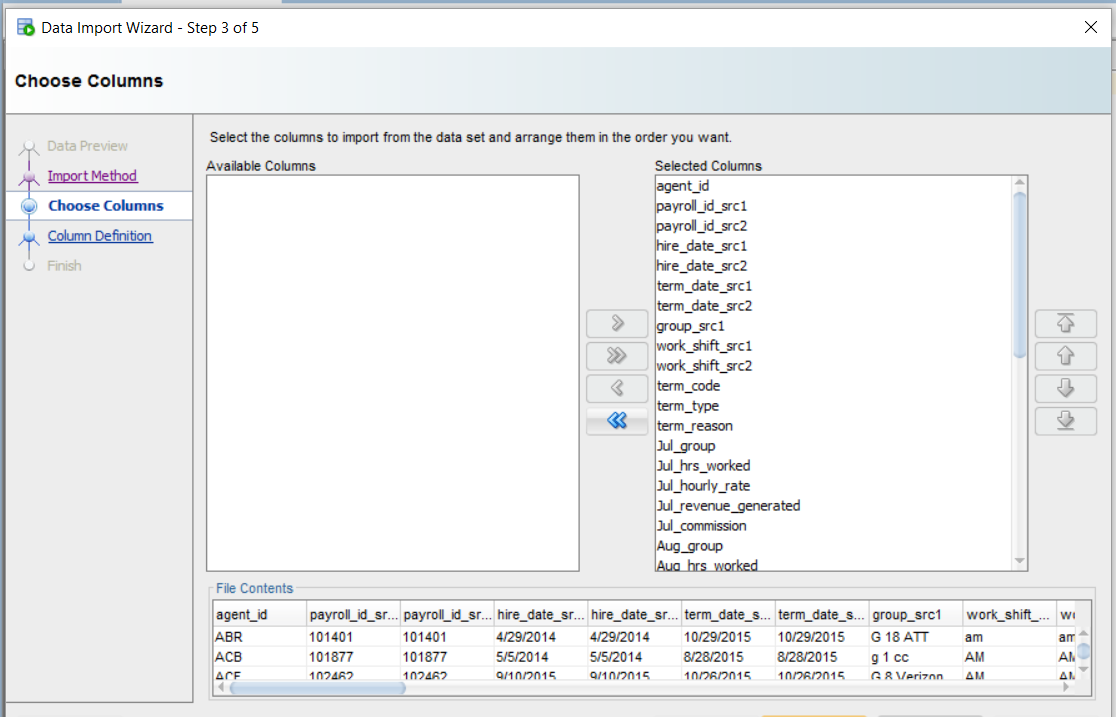
Data import Wizard is used to import the three excel sheets project\_agent\_data, project\_call\_data and project\_feature\_data into Oracle database. Below are the steps followed to achieve this.



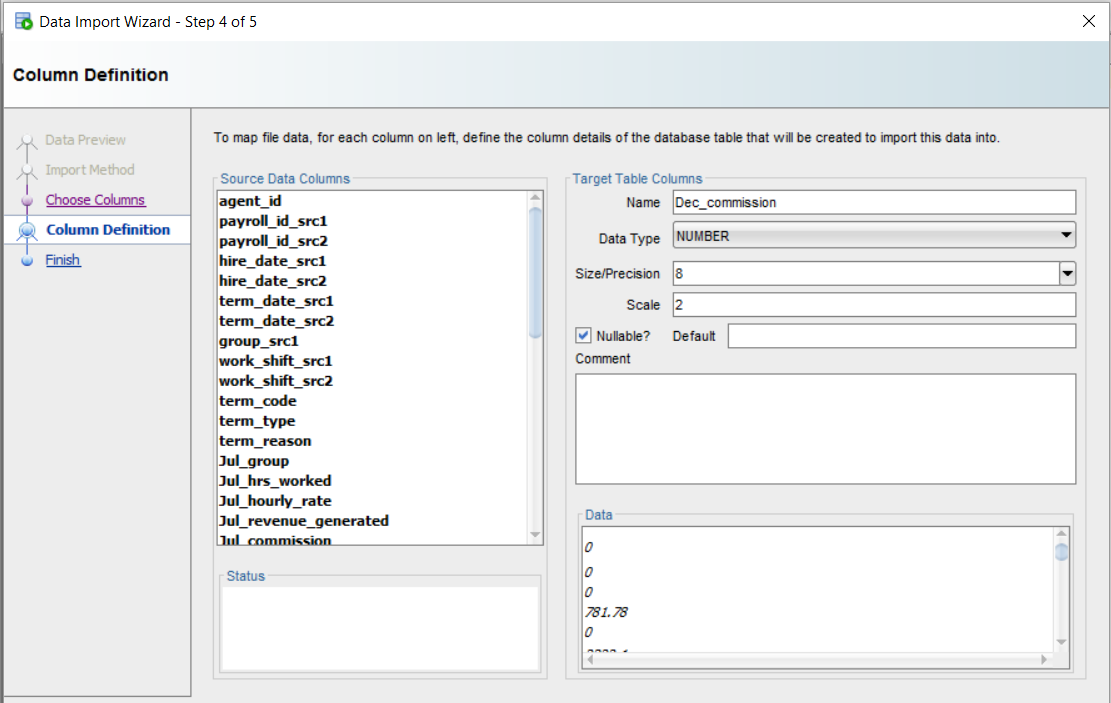
Step 1: Browse and select the excel sheet which has the data.



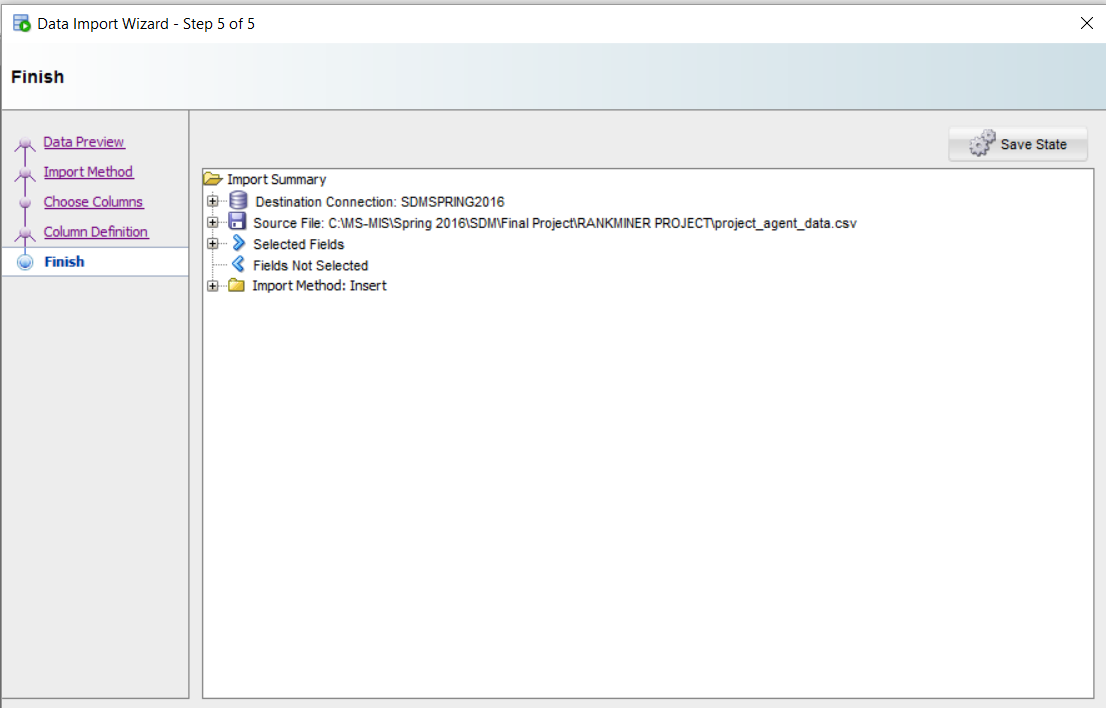
Step 2: Select Insert as the import method. This method creates a new table and inserts data form the excel sheet. Specify the name of the new table.



Step 3: Select the columns of data need to be imported into database.



Step 4: Define the data type for each column and specify the default value if there are any.



Step 5: Review all the details of the import process and click finish to begin the import.

Additionally a column Account is created in Feature\_data table. The values for Account column is extracted from the Audio\_File column. The objective to have this Account column is to establish a relationship with Call\_data on Account Column.

SQL Code to extract create Account column and extract values from Audio\_File column:

CREATE TABLE feature\_data\_temp1

AS (SELECT \* FROM FEATURE\_DATA)

ALTER TABLE FEATURE\_DATA ADD ACCOUNT VARCHAR2(20 BYTE) default null

create table Temp (ACCOUNT VARCHAR2(20 BYTE) default NULL, AUDIO\_FILE\_NAME varchar2(100 byte) NULL)

insert into Temp (ACCOUNT, AUDIO\_FILE\_NAME)(SELECT

SUBSTR(FEATURE\_DATA.AUDIO\_FILE\_NAME,16,8) as ACCOUNT,

FEATURE\_DATA.AUDIO\_FILE\_NAME as AUDIO\_FILE\_NAME

from FEATURE\_DATA )

UPDATE FEATURE\_DATA

SET ACCOUNT =

(SELECT ACCOUNT

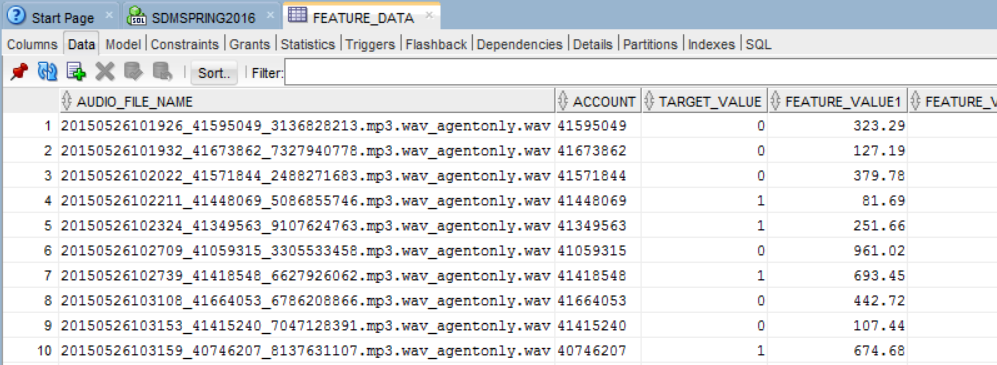
FROM Temp

WHERE Temp.AUDIO\_FILE\_NAME= FEATURE\_DATA.AUDIO\_FILE\_NAME)

drop table feature\_data\_temp

rename Feature\_data to FEATURE\_DATA\_TEMP

rename FEATURE\_DATA\_N to FEATURE\_DATA



Data Cleaning Process:

To begin with, we loaded the dataset in R with all NULL values as “NA” – see below code.

agent <- read.csv('project\_agent\_data.csv',header = TRUE, stringsAsFactors = FALSE)

call <- read.csv('project\_call\_data.csv', header = TRUE, stringsAsFactors = FALSE)

feature <- read.csv('project\_feature\_data.csv',header =TRUE, stringsAsFactors = FALSE)

attach(agent)

attach(call)

attach(feature)

str(agent)

str(call)

str(feature)

agent[agent == ""] <- NA

call[call == ""] <- NA

feature[feature == ""] <- NA

#In the code below, we are using data sources 1 and 2 and merging them for each of the columns with 2 data sources. In case of a discrepancy, for now, we have assumed source 1 to be authentic

agent$payroll\_id\_src1[is.na(agent$payroll\_id\_src1)] <- agent$payroll\_id\_src2[is.na(agent$payroll\_id\_src1)]

agent$payroll\_id\_src2[is.na(agent$payroll\_id\_src2)] <- agent$payroll\_id\_src1[is.na(agent$payroll\_id\_src2)]

agent$hire\_date\_src1[is.na(agent$hire\_date\_src1)] <- agent$hire\_date\_src2[is.na(agent$hire\_date\_src1)]

agent$hire\_date\_src2[is.na(agent$hire\_date\_src2)] <- agent$hire\_date\_src1[is.na(agent$hire\_date\_src2)]

agent$term\_date\_src1[is.na(agent$term\_date\_src2)] <- agent$term\_date\_src1[is.na(agent$term\_date\_src2)]

agent$term\_date\_src1[is.na(agent$term\_date\_src1)] <- agent$term\_date\_src2[is.na(agent$term\_date\_src1)]

agent$work\_shift\_src1[is.na(agent$work\_shift\_src1)] <- agent$work\_shift\_src2[is.na(agent$work\_shift\_src1)]

agent$work\_shift\_src2[is.na(agent$work\_shift\_src2)] <- agent$work\_shift\_src1[is.na(agent$work\_shift\_src2)]

# Check if src1 and src2 are different

#Check Payroll

agent$payroll\_id\_src1[agent$payroll\_id\_src1!=agent$payroll\_id\_src2]

agent$payroll\_id\_src2[agent$payroll\_id\_src1!=agent$payroll\_id\_src2]

#Check Hire Dates

agent$hire\_date\_src1[agent$hire\_date\_src1!=agent$hire\_date\_src2 ]

agent$hire\_date\_src2[agent$hire\_date\_src1!=agent$hire\_date\_src2]

#Check Term Dates

agent$term\_date\_src1[agent$term\_date\_src1!=agent$term\_date\_src2 & !is.na(agent$term\_date\_src1) & !is.na(agent$term\_date\_src2)]

agent$term\_date\_src2[agent$term\_date\_src1!=agent$term\_date\_src2 & !is.na(agent$term\_date\_src1) & !is.na(agent$term\_date\_src2)]

#Check Work Shift

agent$work\_shift\_src1[agent$work\_shift\_src1!=agent$work\_shift\_src2 & !is.na(agent$work\_shift\_src1) & !is.na(agent$work\_shift\_src2)]

agent$work\_shift\_src2[agent$work\_shift\_src1!=agent$work\_shift\_src2 & !is.na(agent$work\_shift\_src1) & !is.na(agent$work\_shift\_src2)]

#Delete src2 for all the dual sourced columns

agent$payroll\_id\_src2 = NULL

agent$hire\_date\_src2 = NULL

agent$term\_date\_src2 = NULL

agent$work\_shift\_src2 = NULL

#Reassign new names to the single new data source created

names(agent)[names(agent)=="payroll\_id\_src1"] <- "payroll\_id"

names(agent)[names(agent)=="hire\_date\_src1"] <- "hire\_date"

names(agent)[names(agent)=="term\_date\_src1"] <- "term\_date"

names(agent)[names(agent)=="work\_shift\_src1"] <- "work\_shift"

names(agent)[names(agent)=="group\_src1"] <- "group"

#Create a binry column for agent terminated or not based on the presence of term\_date

agent$terminated <- 0

agent$terminated[!is.na(agent$term\_date)] <- 1

#We view the below correlations in R, however, we present visualizations for them later

table(agent$terminated)

table(agent$terminated,agent$term\_type)

table(agent$terminated,agent$term\_reason)

round(prop.table(table(agent$terminated[agent$terminated==1], agent$term\_type[agent$terminated==1]), 1)\*100)

#Created a dataset which has only Unique Records from the AGENT DATASET

agent\_unique\_l <- lapply(agent[,c("group","work\_shift","term\_code","term\_type","term\_reason","Jul\_group","Aug\_group","Sep\_group","Oct\_group","Nov\_group","Dec\_group")], unique)

agent\_unique\_max.ln <- max(sapply(agent\_unique\_l, length))

agent\_unique\_l <- lapply(agent\_unique\_l, function(v) { c(v, rep("Z#",

agent\_unique\_max.ln-length(v)))})

agent\_unique <- as.data.frame(agent\_unique\_l)

agent\_unique <- apply(agent\_unique,2,sort,decreasing=F)

names(call)

names(call)[names(call)=="ACCOUNT"] <- "account"

names(call)[names(call)=="AUDIO.FILE.NAME"] <- "audio\_file\_name"

names(call)[names(call)=="SKILL.NAME"] <- "skill\_name"

names(call)[names(call)=="CALL.START.TIME"] <- "call\_start"

names(call)[names(call)=="CALL.END.TIME"] <- "call\_end"

names(call)[names(call)=="AGENT.ID"] <- "agent\_id"

names(call)[names(call)=="CALL.DIRECTION"] <- "call\_direction"

names(call)[names(call)=="CALL.DURATION.HMS"] <- "call\_duration"

names(call)[names(call)=="FILESIZE.KB"] <- "audio\_file\_size"

names(call)[names(call)=="REC.STATUS"] <- "call\_end\_status"

#Created a dataset with all the Unique records form the CALL DATASET

call\_unique\_l <- apply(call[,c("skill\_name","call\_direction","call\_end\_status")],2,sort,decreasing=F)

call\_unique\_l <- lapply(call\_unique\_l, unique)

call\_unique\_max.ln <- max(sapply(call\_unique\_l, length))

call\_unique\_l <- lapply(call\_unique\_l, function(v) { c(v, rep("Z#", call\_unique\_max.ln-length(v)))})

call\_unique <- as.data.frame(call\_unique\_l)

#Setting all characters to uppercase

agent = data.frame(lapply(agent, function(v) {

if (is.character(v)) return(toupper(v))

else return(v)

}))

#102474 to be removed from Dec\_Group

agent$Dec\_group[agent$Dec\_group==102474]=""

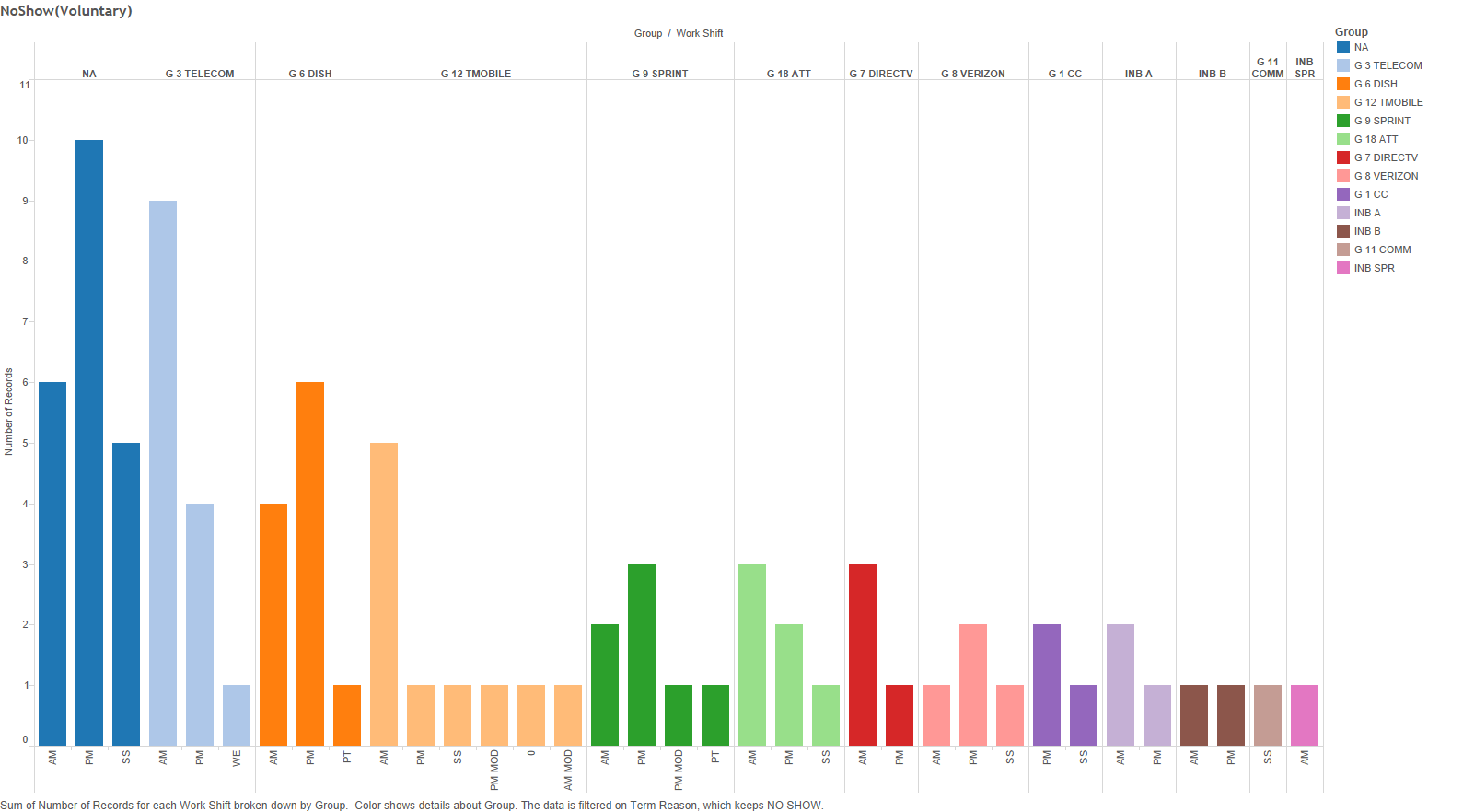
#Converting NA's in term\_type to UNKNOWN for reasons unknown

agent$term\_type[which(!is.na(agent$term\_date) & is.na(agent$term\_type))]="UN"

#Write to csv and export to visualizations

write.csv(agent,"Data\_Visualizaion.csv")

Visualizations based on Cleaned Data:



The above graph depicts the number of employee who voluntarily left the firm without any specific reason.



The above graph shows the count of employees sliced on termination type and group the employee works for.



The above graph provided information on the termination type whether voluntarily or involuntarily of employees working in a particular shift.



The above pie chart gives a bird eye view of the percentage of employees leaving the firm voluntarily or involuntarily.



The above chart presents a drill down on the termination reason for all employees. For example, if an employee leaves voluntarily then for what reason he left.

