Name: Snehanshu Shankar

R Script

```
#Below is my code written in R to generate cleaned data set and export it to excel.

Abandoned_Data_Seed <- read.csv("F:/Softwares/Abandoned_Data_Seed.csv",stringsAsFactors = F,na.strings="")
Reservation_Data_Seed<-read.csv("F:/Softwares/Reservation_Data_Seed.csv",stringsAsFactors = F,na.strings="")

#attach(Reservation_Data_Seed)
attach(Abandoned_Data_Seed)

############Code to remove Duplicates from Abandoned Table##############
Aban_Dupl_EmailRows <-duplicated(Abandoned_Data_Seed$Email,incomparables = NA)
Aban_Dupl_IncPhnRows <-duplicated(Abandoned_Data_Seed$Incoming_Phone,incomparables = NA)
Aban_Dupl_ContPhnRows <-duplicated(Abandoned_Data_Seed$Contact_Phone,incomparables = NA)
Aban_NonDuplicateRows <- !Aban_Dupl_EmailRows & !Aban_Dupl_IncPhnRows & !Aban_Dupl_ContPhnRows
sum(Aban_NonDuplicateRows)
Abandoned_Data_Seed <- Abandoned_Data_Seed[Aban_NonDuplicateRows,]
Abandoned_Data_Seed
nrow(Abandoned_Data_Seed)

#[1] 8297  (Original Rows were 8442, but after removing duplicate tuples on the basis of Email,Incoming_Phone & Contact_Phone, 8297 records in the table are left.)

###################Logic to Differntiate same column from two tables for comparision###############
#As NA in the table gives inconsistent result, therefore, In below mentioned code, I have assigned 1 for NA's in column Contact_Phone, Incoming_Phone & Email in #Abandoned_Data_Seed table and 0 for NA's in column Contact_Phone, Incoming_Phone & Email in #Reservation_Data_Seed table.
Reservation_Data_Seed$Email[is.na(Reservation_Data_Seed$Email)]<-0
Abandoned_Data_Seed$Email[is.na(Abandoned_Data_Seed$Email)]<-1
Reservation_Data_Seed$Contact_Phone[is.na(Reservation_Data_Seed$Contact_Phone)]<-0
Abandoned_Data_Seed$Contact_Phone[is.na(Abandoned_Data_Seed$Contact_Phone)]<-1
Reservation_Data_Seed$Incoming_Phone[is.na(Reservation_Data_Seed$Incoming_Phone)]<-0
Abandoned_Data_Seed$Incoming_Phone[is.na(Abandoned_Data_Seed$Incoming_Phone)]<-1

#Matched Contact_Phone, Incoming_Phone & Email columns in Abandoned_Data_Seed Reservation_Data_Seed tables from Reservation_Data_Seed to check to fetch out people who made reservation.
matchesInPhone = Abandoned_Data_Seed$Incoming_Phone %in% Reservation_Data_Seed$Incoming_Phone
sum(matchesInPhone)

matchesEmail = Abandoned_Data_Seed$Email %in% Reservation_Data_Seed$Email
sum(matchesEmail)

matchesContactPh = Abandoned_Data_Seed$Contact_Phone %in% Reservation_Data_Seed$Contact_Phone
sum(matchesContactPh)

RowsInAbdFromResTb = matchesInPhone | matchesContactPh | matchesEmail
sum(RowsInAbdFromResTb)

ResDataFromAbandoned = Abandoned_Data_Seed[RowsInAbdFromResTb,]
nrow(ResDataFromAbandoned)


##############Tested below mentioned Logic to Remove duplicate rows except NA from below Sample DF##
```

```
employee <- c('John Doe','Peter Gynn','Jolie Hope',NA,NA,'John Doe','John Doe')
salary <- c(21000, 23400, 26800,NA,NA,21000,NA)
employ.data <- data.frame(employee, salary)
data <-duplicated(employ.data$employee,incomparables = NA)
employ.data[!data,]


#Now merge rows like rbind but remove duplicate rows(rbind not used as it does not remove dulplicate tuples)
employee <- c('A',NA,NA,'A','B')
salary <- c(6,3,4,5,6)
employ.data <- data.frame(employee, salary)
em_em<-duplicated(employ.data$employee,incomparables = NA)
em_sal<-duplicated(employ.data$salary,incomparables = NA)
uniqueRows <- !em_em & !em_sal
employ.data[uniqueRows,]


######################################################################

Reservation_Data_Seed$Email[is.na(Reservation_Data_Seed$Email)]<-1
Abandoned_Data_Seed$Email[is.na(Abandoned_Data_Seed$Email)]<-0
Reservation_Data_Seed$Contact_Phone[is.na(Reservation_Data_Seed$Contact_Phone)]<-1
Abandoned_Data_Seed$Contact_Phone[is.na(Abandoned_Data_Seed$Contact_Phone)]<-0
Reservation_Data_Seed$Incoming_Phone[is.na(Reservation_Data_Seed$Incoming_Phone)]<-1
Abandoned_Data_Seed$Incoming_Phone[is.na(Abandoned_Data_Seed$Incoming_Phone)]<-0


  matchesInPhone = Abandoned_Data_Seed$Incoming_Phone %in% Reservation_Data_Seed$Incoming_Phone
  sum(matchesInPhone)
  matchesEmail = Abandoned_Data_Seed$Email %in% Reservation_Data_Seed$Email
  sum(matchesEmail)
  matchesContactPh = Abandoned_Data_Seed$Contact_Phone %in% Reservation_Data_Seed$Contact_Phone
  sum(matchesContactPh)
  RowsInAbdFromResTb = matchesInPhone | matchesContactPh | matchesEmail
  sum(RowsInAbdFromResTb)
  ResDataFromAbandoned = Abandoned_Data_Seed[RowsInAbdFromResTb,]
  nrow(ResDataFromAbandoned)
  ############Session Calculation##########
  ResDataFromAbandoned["Reservation_Session"]<-NA
  ContactphoneMatchIndex<-
match(ResDataFromAbandoned$Contact_Phone,Reservation_Data_Seed$Contact_Phone,nomatch = 0)
  IncomingPhoneMatchIndex<-
match(ResDataFromAbandoned$Incoming_Phone,Reservation_Data_Seed$Incoming_Phone,nomatch = 0)
  EmailMatchIndex<- match(ResDataFromAbandoned$Email,Reservation_Data_Seed$Email,nomatch = 0)
  ResDataFromAbandoned$Reservation_Session<-
Reservation_Data_Seed$Session[(IncomingPhoneMatchIndex[[2]])]

  for(i in 1:nrow(ResDataFromAbandoned)){
    if(ContactphoneMatchIndex[i]!=0){
      ResDataFromAbandoned$Reservation_Session[i]                                    <-
Reservation_Data_Seed$Session[ContactphoneMatchIndex[i]]
    }else if(IncomingPhoneMatchIndex[i]!=0){
      ResDataFromAbandoned$Reservation_Session[i]                                    <-
Reservation_Data_Seed$Session[IncomingPhoneMatchIndex[i]]
    }else if(EmailMatchIndex!=0){
      ResDataFromAbandoned$Reservation_Session[i] <- Reservation_Data_Seed$Session[EmailMatchIndex[i]]
    }
```

```
  }
  ###Days In Between###
  ResDataFromAbandoned["Days_in_Between"]<-0
  ResDataFromAbandoned$Days_in_Between<-
as.numeric(as.Date(ResDataFromAbandoned$Reservation_Session, "%Y.%m.%d %H:%M:%S") -
as.Date(ResDataFromAbandoned$Session, "%Y.%m.%d %H:%M:%S"))

  ##Days In Between Abandoned Data Set##
  Abandoned_Data_Seed["Days_in_Between"]<-NA
  matchCaller4DaysInBetwIndx <-
match(Abandoned_Data_Seed$Caller_ID,ResDataFromAbandoned$Caller_ID,nomatch = 0)
  Abandoned_Data_Seed["Days_in_Between"]<-0
  for(i in 1:nrow(Abandoned_Data_Seed)){
    if(matchCaller4DaysInBetwIndx[i]!=0){
      Abandoned_Data_Seed$Days_in_Between[i]<-
ResDataFromAbandoned$Days_in_Between[matchCaller4DaysInBetwIndx[i]]
    }else{
      Abandoned_Data_Seed$Days_in_Between[i]<-200
    }
  }

  ##CustomerID
  #Abandoned_Data_Seed["CustomerID"]<-NULL
  Abandoned_Data_Seed["CustomerID"]<- c(1:nrow(Abandoned_Data_Seed))

  ##New data frame to export excel
  CleanedAbandonedData <-
data.frame(Abandoned_Data_Seed$CustomerID,Abandoned_Data_Seed$Test_Variable,Abandoned_Data_Seed
$Outcome,Abandoned_Data_Seed$Days_in_Between,Abandoned_Data_Seed$D_State,Abandoned_Data_Seed
$D_Email)
  colnames(CleanedAbandonedData)<-
c("CustomerID","Test_Variable","Outcome","Days_in_Between","D_State","D_Email")

  # Write the Cleaned data set in a Excel file.
  write.xlsx(CleanedAbandonedData, file="E:/CleanedData.xlsx",sheetName="CleanedData", append=FALSE)
  #write.xlsx(CleanedAbandonedData, "E:/CleanedData.xlsx")

  model=lm(CleanedAbandonedData$Outcome~CleanedAbandonedData$Test_Variable)
  summary(model)
  ##Q12 A/B Testing

###To Map outcome with Abandon dataset
Abandoned_Data_Seed["Outcome"]<-NA
OutcomeVector = Abandoned_Data_Seed$Caller_ID %in% ResDataFromAbandoned$Caller_ID
OutcomeVector
OutcomeVector<-as.integer(OutcomeVector)
Abandoned_Data_Seed$Outcome<-OutcomeVector
Cust_in_Test_NotPurchased<-subset(Abandoned_Data_Seed,  Abandoned_Data_Seed$Test_Control=="test"  &
Abandoned_Data_Seed$Outcome==0 )
Cust_in_Test_NotPurchased
nrow(Cust_in_Test_NotPurchased)

Cust_in_Test_Purchased<-subset(Abandoned_Data_Seed,    Abandoned_Data_Seed$Test_Control=="test"    &
Abandoned_Data_Seed$Outcome==1 )
Cust_in_Test_Purchased
nrow(Cust_in_Test_Purchased)
```

```
Cust_in_Control_NotPurchased<-subset(Abandoned_Data_Seed,
Abandoned_Data_Seed$Test_Control=="control" & Abandoned_Data_Seed$Outcome==0 )
Cust_in_Control_NotPurchased
nrow(Cust_in_Control_NotPurchased)

Cust_in_Control_Purchased<-subset(Abandoned_Data_Seed, Abandoned_Data_Seed$Test_Control=="control" &
Abandoned_Data_Seed$Outcome==1 )
Cust_in_Control_Purchased
nrow(Cust_in_Control_Purchased)

############### + state
Cust_in_Test_NotPurchased_AK<-subset(Abandoned_Data_Seed,  Abandoned_Data_Seed$Test_Control=="test"
& Abandoned_Data_Seed$Outcome==0 & Abandoned_Data_Seed$Address=="WI")
Cust_in_Test_NotPurchased_AK
nrow(Cust_in_Test_NotPurchased_AK)

Cust_in_Test_Purchased_AK<-subset(Abandoned_Data_Seed,  Abandoned_Data_Seed$Test_Control=="test"  &
Abandoned_Data_Seed$Outcome==1 & Abandoned_Data_Seed$Address=="WI")
Cust_in_Test_Purchased_AK
nrow(Cust_in_Test_Purchased_AK)

Cust_in_Control_NotPurchased_AK<-subset(Abandoned_Data_Seed,
Abandoned_Data_Seed$Test_Control=="control"  &  Abandoned_Data_Seed$Outcome==0  &
Abandoned_Data_Seed$Address=="WI")
Cust_in_Control_NotPurchased_AK
nrow(Cust_in_Control_NotPurchased_AK)

Cust_in_Control_Purchased_AK<-subset(Abandoned_Data_Seed,
Abandoned_Data_Seed$Test_Control=="control"  &  Abandoned_Data_Seed$Outcome==1  &
Abandoned_Data_Seed$Address=="WI")
Cust_in_Control_Purchased_AK
nrow(Cust_in_Control_Purchased_AK)


############Adding Test Variable
#Abandoned_Data_Seed["Test Variable"]<-NULL
Abandoned_Data_Seed$Test_Variable[Abandoned_Data_Seed$Test_Control=='test']<-1
Abandoned_Data_Seed$Test_Variable[Abandoned_Data_Seed$Test_Control=='control']<-0

############Adding D_State Variable
#Abandoned_Data_Seed["D_State"]<-NULL
Abandoned_Data_Seed$D_State[Abandoned_Data_Seed$Address!=""]<-1
Abandoned_Data_Seed$D_State[is.na(Abandoned_Data_Seed$Address)]<-0

############Adding D_Email Variable
#Abandoned_Data_Seed["D_Email"]<-NULL
Abandoned_Data_Seed$D_Email[Abandoned_Data_Seed$Email!=0]<-1
Abandoned_Data_Seed$D_Email[Abandoned_Data_Seed$Email==0]<-0

##Adding INT_TV_DState
CleanedAbandonedData["INT_TV_DState"]<-
CleanedAbandonedData$Test_Variable*CleanedAbandonedData$D_State
##Adding INT_TV_DEmail
CleanedAbandonedData["INT_TV_DEmail"]<-
CleanedAbandonedData$Test_Variable*CleanedAbandonedData$D_Email
```

```
#Q12
CustInControlGrp<-subset(CleanedAbandonedData,CleanedAbandonedData$Test_Variable==0)
CusInTestGrp<-subset(CleanedAbandonedData,CleanedAbandonedData$Test_Variable==1)

t.test(CusInTestGrp$Outcome,CustInControlGrp$Outcome,alternative='greater')
#Q14

model<- lm(CleanedAbandonedData$Outcome~CleanedAbandonedData$Test_Variable +
CleanedAbandonedData$D_State +CleanedAbandonedData$D_Email)
summary(model)

model<- lm(CleanedAbandonedData$Outcome~CleanedAbandonedData$Test_Variable +
CleanedAbandonedData$D_State +CleanedAbandonedData$D_Email +CleanedAbandonedData$INT_TV_DState
+CleanedAbandonedData$INT_TV_DEmail)
summary(model)

#Q15

Case1:
ResDataFromAbandoned["Test_Variable"]<-NA
ResDataFromAbandoned$Test_Variable[ResDataFromAbandoned$Test_Control=='test']<-1
ResDataFromAbandoned$Test_Variable[ResDataFromAbandoned$Test_Control=='control']<-0

model<- lm(ResDataFromAbandoned$Days_in_Between ~ ResDataFromAbandoned$Test_Variable)
summary(model)

Case 2:
Outcome_vs_DaysInBet <- subset(CleanedAbandonedData, CleanedAbandonedData$Outcome==1)
model<- lm(Outcome_vs_DaysInBet$Outcome ~ Outcome_vs_DaysInBet$Days_in_Between)
summary(model)

#PurchasedData<- subset(CleanedAbandonedData, CleanedAbandonedData$Outcome==1)
#str(ResDataFromAbandoned)
###Random state
sample(Abandoned_Data_Seed$Address, 5, replace = FALSE)

##library(xlsx)
##write.xlsx(Abandoned_Data_Seed_Final, "c:/Abandoned_Data_Seed_Final.xlsx")
########################Segregating Control & Test Group from Abandoned Table###################
ResDataControlGrp = subset(Customers_Purchased_Data_, Customers_Purchased_Data_$Test_Control=="test"
)
nrow(ResDataControlGrp)
ResDataTestGrp =  subset(Customers_Purchased_Data_, Customers_Purchased_Data_$Test_Control=="control"
)
nrow(ResDataTestGrp)
install.packages("xlsx")
library(xlsx)
# Write the first data set in a new workbook
write.xlsx(Abandoned_Data_Seed_Final, file="C:\testCleanedData.xlsx",sheetName="CleanedData",
append=FALSE)
```